

**AUTENTIKASI BIOMETRIK BERBASIS *FACE RECOGNITION*
MENGUNAKAN METODE *CONVOLUTIONAL NEURAL*
NETWORK UNTUK SIMULASI *BARRIER GATE SYSTEM***

SKRIPSI



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun Oleh:

**MOCHAMMAD LANGGENG PRASETYO
H76216062**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA
2020**

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini,

Nama : Mochammad Langgeng Prasetyo

NIM : H76216062

Program Studi : Sistem Informasi

Angkatan : 2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: “AUTENTIKASI BIOMETRIK BERBASIS *FACE RECOGNITION* MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* UNTUK SIMULASI *BARRIER GATE SYSTEM*”. Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 5 Agustus 2020

Yang menyatakan,



(Mochammad Langgeng Prasetyo)

NIM. H76216062

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi Oleh

NAMA : MOCHAMMAD LANGGENG PRASETYO

NIM : H76216062

JUDUL : AUTENTIKASI BIOMETRIK BERBASIS *FACE*
RECOGNITION MENGGUNAKAN METODE
CONVOLUTIONAL NEURAL NETWORK UNTUK
SIMULASI *BARRIER GATE SYSTEM*

Ini telah diperiksa dan disetujui untuk diujikan.

Surabaya, 01 Juli 2020

Dosen Pembimbing 1



(Achmad Teguh Wibowo, MT)
NIP 198810262014031003

Dosen Pembimbing 2



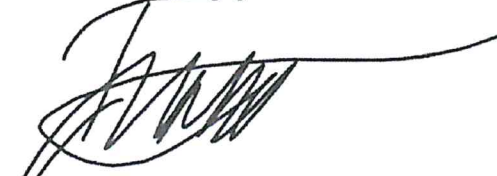
(Mujib Ridwan, S.Kom., M.T)
NIP 198604272014031004

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Mochammad Langgeng Prasetyo ini telah dipertahankan
di depan tim penguji skripsi
di Surabaya, 06 Juli 2020.

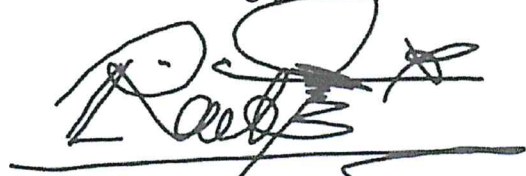
Mengesahkan,
Dewan Penguji,

Penguji 1



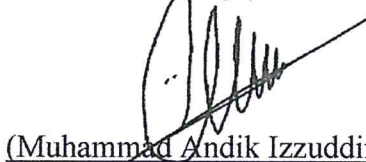
(Achmad Teguh Wibowo, MT)
NIP 198810262014031003

Penguji 2



(Mujib Ridwan, S.Kom., M.T)
NIP 198604272014031004

Penguji 3



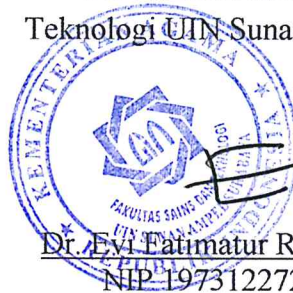
(Muhammad Andik Izzuddin, MT)
NIP 198403072014031001

Penguji 4



(Ahmad Yusuf, M.Kom)
NIP 199001202014031003

Mengetahui,
Plt. Dekan Fakultas Sains dan
Teknologi UIN Sunan Ampel Surabaya



Dr. Evi Fatimatur Rusydiyah, M.Ag.
NIP 197312272005012003



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : MOCHAMMAD LANGGENG PRASETYO
NIM : H76216062
Fakultas/Jurusan : SAINS DAN TEKNOLOGI / SISTEM INFORMASI
E-mail address : langgengprasetyo.911@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

☒ Sekripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

AUTENTIKASI BIOMETRIK BERBASIS FACE RECOGNITION MENGGUNAKAN

METODE CONVOLUTIONAL NEURAL NETWORK UNTUK

SIMULASI BARRIER GATE SYSTEM

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara **fulltext** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya,

Penulis

(MOCHAMMAD LANGGENG PRASETYO)

ABSTRAK

AUTENTIKASI BIOMETRIK BERBASIS FACE RECOGNITION MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK UNTUK SIMULASI BARRIER GATE SYSTEM

Oleh:

Mochammad Langgeng Prasetyo

Wajah adalah karakteristik unik yang dimiliki oleh masing-masing individu. Keunikan ini dapat digunakan sebagai otentikasi dan identifikasi yang dapat diimplementasikan untuk keamanan, transaksi, otentikasi. Gambar wajah masing-masing individu adalah objek yang kondisinya susah dikontrol. Penyebabnya adalah gambar wajah dapat berubah dengan berbagai posisi dan tingkat kemiringan kepala, posisi yang berbeda, dan perbedaan intensitas cahaya, yang membuat proses pengenalan wajah sulit diklasifikasi. Dalam penelitian ini membahas tentang otentikasi biometrik berbasis *face recognition* untuk simulasi *barrier gate system* menggunakan algoritma convolutional neural network berdasarkan jaringan saraf tiruan untuk melakukan klasifikasi gambar secara real-time. Proses ini terdiri dari *convolutional layer*, *pooling layer*, *max pooling*, *flattening*, dan *fully connected layer* untuk mendeteksi wajah. *Output* dari proses sebelumnya dikirim ke mikrokontroler berbasis teknologi IoT. Evaluasi hasil penelitian dengan data 100 mendapat tingkat kesalahan rata-rata 0,3205, tingkat keberhasilan sistem 94%, dan waktu respon rata yang diperlukan oleh mikrokontroler 0,56217634 ms serta hasil evaluasi akurasi sistem dalam model confusion matrix sebesar 93,3%. Algoritma CNN dalam penerapan *face recognition* mampu menghasilkan nilai *error* yang rendah, tingkat keberhasilan dan akurasi tinggi, serta cepat dalam mengidentifikasi citra wajah.

Kata kunci : *Face Recognition, Autentikasi, Identification, Extended Local Binary Pattern, Convolutional Neural Network, Barrier Gate System.*

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN	i
LEMBAR PERSETUJUAN PEMBIMBING	iii
LEMBAR PENGESAHAN TIM PENGUJI.....	iii
MOTTO	iv
KATA PENGANTAR.....	v
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1. Tinjauan Penelitian Terdahulu.....	6
2.2. Dasar Teori	8
2.2.1. <i>Human Computer Interaction</i>	8
2.2.2. <i>Biometrics</i>	9
2.2.3. <i>Computer Vision</i>	9
2.2.4. <i>Citra Digital</i>	10
2.2.5. <i>Machine Learning</i>	10
2.2.6. <i>Deep Learning</i>	11
2.2.7. <i>Neural Network</i>	12
2.2.8. <i>Convolutional Neural Network</i>	13
2.2.9. <i>Face Recognition</i>	27
2.2.10. <i>Face Expression</i>	29
2.2.11. <i>Model Waterfall</i>	29
2.2.12. <i>Unified Modeling Language (UML)</i>	30
2.2.13. <i>Barrier Gate System</i>	31
2.2.14. <i>Web Application</i>	32

DAFTAR TABEL

Tabel 3. 1 Jenis Pengumpulan Data	38
Tabel 3. 2 Pengujian Math Static	42
Tabel 3. 3 Detil Rata Rata Waktu	42
Tabel 3. 4 Konsep ROC (Receive Operating Characteristic)	43
Tabel 3. 5 Tabel Pengujian ROC	44
Tabel 4. 1 Kebutuhan Hardware	47
Tabel 4. 2 Percobaan Face Recognition.....	88
Tabel 4. 3 Pengujian Math Static	96
Tabel 4. 4 Assessment Pengujian Math Static	100
Tabel 4. 5 Skenario Pengujian ROC	101
Tabel 4. 6 Pengujian ROC True Value	102
Tabel 4. 7 Hasil ROC True Condition Face Identification	105
Tabel 4. 8 Hasil ROC True Condition Face Expression	106
Tabel 4. 9 Pengujian ROC False Value	106
Tabel 4. 10 Hasil ROC False Condition Face Identification	108
Tabel 4. 11 Hasil ROC False Condition Face Expression	110
Tabel 4. 12 Evaluasi Hasil Keseluruhan	111
Tabel 4. 13 Skenario Pengujian Akurasi.....	114
Tabel 4. 14 Skenario Pertama	115
Tabel 4. 15 Skenario Kedua.....	116
Tabel 4. 16 Skenario ketiga.....	116
Tabel 4. 17 Skenario keempat.....	117
Tabel 4. 18 Skenario Kelima.....	118
Tabel 4. 19 Skenario Keenam	119
Tabel 4. 20 Pengelompokan Persentase Akurasi	119
Tabel 4. 21 Hasil Skenario Keseluruhan.....	120

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan informasi dan teknologi pada era digitalisasi, berdampak pada terintegrasinya penggunaan *Information Technology* (TI) secara global. Dapat terjadi karena perkembangan zaman sesuai dengan kemajuan teknologi (Gangopadhyay, 2018). Kemajuan teknologi dan ilmu pengetahuan semakin pesat dapat kita temukan dalam kehidupan sehari-hari seperti *face recognition* dengan *input citra digital*, pemanfaatan *fingerprint* untuk keamanan perangkat *mobile*, *voice recognition* untuk deteksi suara dapat membantu dalam penulisan *text*, dan teknologi lain sebagainya (Bah & Ming, 2019). Salah satu penerapan teknologi yang digunakan seperti autentikasi menggunakan citra wajah digital dengan mengintegrasikan *face recognition* dan *face expression*. *Face recognition* bertujuan untuk mendeteksi suatu objek dengan melakukan *scanning data* dari data *training*, sedangkan *face expression* bertujuan untuk mengetahui emosi dari objek yang sedang dideteksi (Revina & Emmanuel, 2018).

Sistem autentikasi biometrik yang memiliki akurasi tinggi dan memiliki tingkat *error* rendah perlu dikembangkan, dilatarbelakangi dengan meningkatnya tingkat kejahatan dan kebutuhan *access control* untuk memonitor perilaku (Setiadi et al., 2017). *Access control* yang dimaksud berupa pengontrolan keluar masuk pengendara dengan model sistem parkir konvensional dapat memicu kesalahan dalam penulisan karcis oleh petugas parkir sehingga membuat antrian kendaraan semakin panjang, serta merugikan pengendara karena karcis yang diperoleh tidak sesuai dengan kendaraannya (Setiadi et al., 2017) (Sujatha et al., 2016). Penanggulangan yang dapat digunakan untuk mengatasi permasalahan tersebut yaitu *authentication* menggunakan *face recognition* untuk simulasi *barrier gate*, penggunaan ini karena wajah merupakan *instrument* yang bersifat unik, yang dimaksud adalah wajah mempunyai ciri khusus yang tidak dimiliki oleh setiap individu dan wajah mempunyai perbedaan yang paling tinggi .

serta sesuai dalam penggunaan sistem *face recognition*. Kemudian dapat digunakan sebagai alat untuk mengidentifikasi pengendara (Athale et al., 2015).

Komponen struktural wajah mempunyai variabilitas yang sangat banyak dan berbeda beda. termasuk bentuk wajah, warna kulit, dan ukuran citra wajah dari setiap manusia. Faktor yang mempengaruhi tingkat akurasi adalah ekspresi wajah seperti tersenyum, sedih, tertawa, marah, terkejut, takut, dan datar. Ekspresi tersebut dapat mempengaruhi tingkat kesulitan dalam proses pengembangannya (Juneja, 2017). Metode *Viola-Jones* merupakan salah satu metode yang digunakan dalam penerapan *face recognition*, namun penggunaan metode ini tidak dapat mengklasifikasi citra jika posisi wajah tidak lurus secara frontal. Jika citra wajah teridentifikasi miring, sangat mempengaruhi tingkat keberhasilan sistem (Putro, M Dwisnanto, 2012).

eksternal maupun internal yang terdapat melalui jaringan tersebut (Nalepa et al., 2020).

Teknik penyempurnaan proses klasifikasi untuk mengurangi parameter bebas dapat menggunakan algoritma *Convolutional Neural Network* (CNN) yang terinspirasi oleh korteks mamalia visual sel sederhana dan kompleks. Model ini dapat mengurangi sejumlah parameter bebas dan dapat menyelesaikan masalah parameter yang berlebihan serta dapat menangani deformasi citra gambar. Dari pembahasan diatas yang menjadi pembeda dari penelitian ini yaitu mengintegrasikan aplikasi biometrik berbasis *face recognition* dengan algoritma CNN dengan perangkat *microcontroller* untuk simulasi *barrier gate system* secara *real-time*. Dan dapat menjadi sebuah inovasi baru untuk penggantian model *barrier gate system* konvensional (Shustanov & Yakimov, 2017).

1.2 Perumusan Masalah

1.3 Batasan Masalah

Batasan masalah merupakan batas pada suatu penelitian agar tetap tertuju pada pembahasan yang sedang dilakukan, batasan tersebut yaitu:

1. Pendeteksian citra wajah harus menghadap kamera dan tidak diwakilkan.
2. Perangkat yang digunakan harus tersedia *camera* dengan minimum spesifikasi *8 mega pixels* 30 Fps dan *resolution* 640 x 480 untuk melakukan pengambilan citra wajah secara *real-time*.
3. Area penempatan kamera harus tersedia intensitas pencahayaan yang terang untuk melakukan proses perekaman citra wajah gambar.
4. Aplikasi yang dibuat hanya berbasis perangkat *Web* dan mempunyai Spesifikasi yang memadai dalam melakukan pengoperasian sistem.
5. Perekaman citra wajah efektif untuk penggunaan 1 Citra wajah dan menggunakan ekspresi tersenyum.
6. Dalam pengembangan sistem berbasis *face recognition* ini tidak difokuskan untuk mengidentifikasi individu kembar identik (serupa).

1.4 Tujuan Penelitian

1. Membuat sistem autentikasi biometrik berbasis *face recognition* menggunakan metode *Convolutional Neural Network*.
2. Mengukur kinerja metode CNN dalam sistem autentikasi biometrik berbasis *face recognition* untuk simulasi *barrier gate system*.

1.5 Manfaat Penelitian

1.5.1 Manfaat Akademik

1. Memberi gambaran mengenai kemampuan mahasiswa dalam menguasai materi dan bidang ilmu yang diajarkan dan diperoleh selama proses perkuliahan.
2. Memberi gambaran mengenai kemampuan mahasiswa dalam penerapan ilmu dan sebagai bahan evaluasi akademik.

1.5.2 Manfaat Aplikatif

1. Dengan adanya pengimplementasian *Face Recognition* ini, dapat dijadikan sebagai pertimbangan dan inovasi dalam hal penggantian model *barrier gate system*.

penggabungan antara HOG (*Histogram of Oriented Gradient*) dengan SVM (*Support Vector Machine*) dan mendapatkan hasil dapat mendeteksi meskipun dalam keadaan *blur*, dapat mendeteksi meskipun dalam ekspresi, pose wajah yang bermacam-macam, dan tetap menghasilkan tingkat keakuratan yang tinggi.

Penelitian yang dilakukan oleh (Jie Wang and Zihao Li, 2018) dari “*School of Electrical Engineering, Zhengzhou University, Zhengzhou, China*” dengan judul “Research On Face Recognition Based on CNN” yang berfokus pada pengembangan *Deep Learning* berbasis algoritma CNN dan melakukan penelitian mengenai Identifikasi Citra wajah berbasis Algoritma CNN yang mempunyai hasil bahwa Algoritma ini bagian dari proses Face Recognition yang digunakan untuk melakukan training data dan untuk mengoptimasi performa sistem.

Penelitian yang dilakukan oleh (Halprin Abhirawa , Jondri,M.Si. , Anditya Arifianto, S.T.,M.T., 2017) dari “Universitas Telkom Indonesia Fakultas Informatika, Prodi S1 Teknik Informatika” dengan judul “Pengenalan Wajah Menggunakan Convolutional Neural Network” berfokus pada Pengembangan sistem pengenalan citra wajah dengan metode CNN dan mendapatkan hasil bahwa Pengembangan sistem ini diimplementasikan terhadap *Data Testing The Extended Yale Face Database* dengan akurasi sebesar 75.79% , sistem yang menggunakan *Dropout* menghasilkan performansi terhadap Data Validasi yang lebih baik dibandingkan dengan sistem yang tidak menggunakan *Dropout*, jika sistem menggunakan *Dropout* mendapatkan akurasi 86,71% dengan selisih akurasi 15,00% terhadap sistem yang tidak menggunakan *Dropout* yang memiliki akurasi 71.71%.

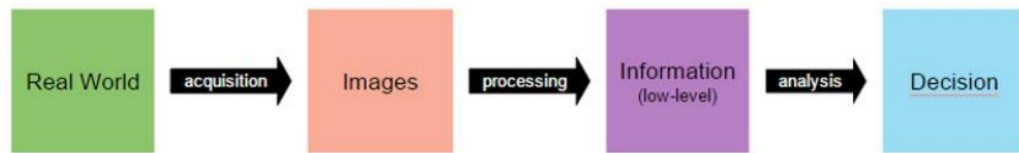
Penelitian yang dilakukan oleh (I Wayan Suartika E. P, Arya Yudhi Wijaya, dan Rully Soelaiman ,2016) dari “Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)” dengan judul penelitian “Klasifikasi Citra Menggunakan *Convolutional Neural Network (CNN)* pada Caltech 101” yang berfokus pada pengembangan aplikasi menggunakan algoritma CNN dan mengevaluasi hasil performa sistem seperti *accuracy, prediction, error rate*, serta melakukan langkah – langkah CNN dari awal hingga akhir, dan mendapatkan hasil bahwa evaluasi performa sistem yaitu mendapatkan hasil akurasi

sebesar 20% - 50 % , perubahan tingkat confusion tidak mempengaruhi hasil akurasi. Hal ini membuktikan bahwa klasifikasi menggunakan algoritma CNN relative handal terhadap perubahan parameter yang dilakukan.

2.2.Dasar Teori

Human Computer Interaction (HCI) pengertian sederhananya dan mudah dipahami oleh masyarakat umum yaitu interaksi atau kolaborasi antara manusia dan *computer*, dimana dalam interaksi tersebut terjadi hubungan timbal balik secara informatif dan interaktif (Gaouar et al., 2018). HCI menurut jurnal merupakan disiplin ilmu pengetahuan berbasis *Technology* yang berfokus pada mempelajari interaksi yang sedang terjadi antara manusia dan komputer, untuk menghasilkan *software* yang fungsional, mempunyai tingkat keamanan tinggi, efektif, serta efisien (Sun et al., 2020).

cara mengambil informasi yang terdapat pada objek tersebut (Tian, Hongkung., 2020). Berikut gambaran proses *Computer Vision*.



Gambar 2.2 *Computer Vision* (Tian, Hongkong., 2020).

Pada gambar 2.2 menggambarkan bahwa proses *computer vision* berawal dari dunia nyata yang kemudian dikonversi menjadi *citra* dan menghasilkan informasi. Tujuan dari *computer vision* adalah untuk meniru *vision* dari dunia nyata ke dalam sistem *computer* secara otomatis menggunakan gambar digital melalui tiga komponen pemrosesan utama yang dijalankan satu persatu yaitu Pengambilan gambar, pemrosesan gambar, analisis dan pemahaman gambar (Feng et al., 2019).

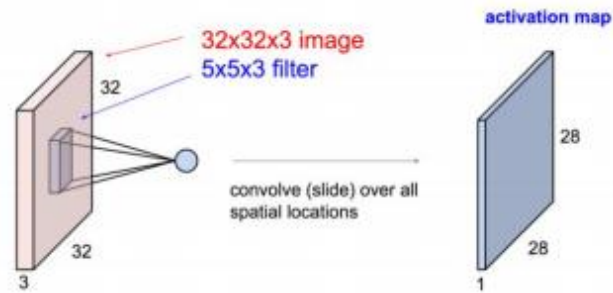
2.2.4. Citra Digital

Citra *digital* adalah gambar/foto 2 dimensi yang dapat diolah dan diukur menggunakan teknologi komputer yang terdiri dari berbagai titik-titik yang dinamakan piksel (pixel). Setiap piksel digambarkan dengan satu kotak kecil (Kadir & Susanto, 2012). Dalam tinjauan dan pengukuran secara matematis, citra merupakan fungsi berkelanjutan dari intensitas cahaya pada bidang dua dimensi.

2.2.5. Machine Learning

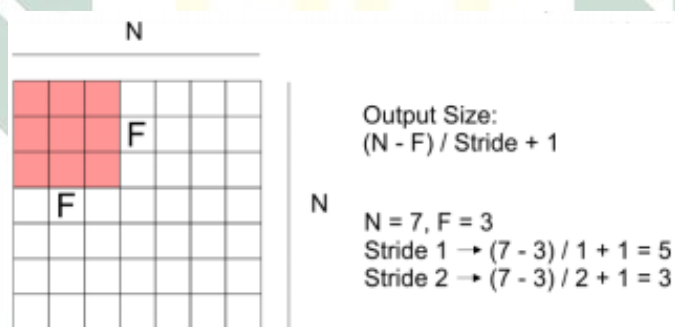
Machine Learning dapat didefinisikan sebagai salah satu dari ilmu kecerdasan buatan manusia yang berfokus dan berdominasi untuk keperluan pengembangan serta studi sistem agar mampu melakukan identifikasi dari data-data yang diperolehnya (Schrouff et al., 2019). *Machine Learning* telah berkontribusi dengan berbagai sektor seperti pengamanan dalam hal *Cyber Security* (Gibert et al., 2020). Cara agar dapat mengaplikasikan teknik-teknik *Machine Learning*, harus terdapat data mentah untuk proses pengenalan. Tanpa adanya data tersebut maka model implementasi *Machine Learning* tidak mampu untuk diaplikasikan. Data yang digunakan biasanya terbagi menjadi dua bagian, yaitu data *training* dan data *testing* yang keduanya mempunyai fungsi yang berbeda. Data *training* biasa digunakan

parameter yaitu *depth*, *stride*, *zero padding*. Berikut konsep dari *convolutional layer*.



Gambar 2.7 Konsep *Filtering* (Santoso & Ariyanto, 2018).

Pada gambar 2.7. Menggambarkan bahwa dari proses *input citra digital* yang berukuran panjang, lebar, dan tinggi 32x32x3, kemudian akan melakukan proses konvolusi (*filtering*) dengan ukuran 5x5x3, proses tersebut menghasilkan *activation map* (Abhirawan et al., 2017). Proses *filtering* tersebut dilakukan dengan rumus seperti gambar berikut.



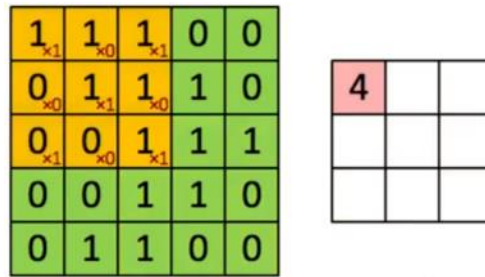
Gambar 2.8 Proses Perhitungan *Binary* (Santoso & Ariyanto, 2018).

Gambar 2.8 Menggambarkan konsep perhitungan pada proses *filtering* dengan inputan *citra image digital* berupa *binary*. N merupakan panjang dan lebar dari *input citra digital*, F merupakan proses *filtering*. Cara pengambilan *binary* dengan cara $(N - F) / \text{Stride} + 1$, dimana teknik pengambilan *matrix* F, total jumlah $N = 7$ dan total jumlah $F = 3$, untuk penggambaran dari $\text{Stride} + 1$, jika proses *matrix* 3x3 sudah selesai maka akan bergeser pada 1 *pixel* sebelahnnya, maka akan menjadi *matrix* seperti pada gambar 2.8. Berikut detil pengambilan *binary* dan proses perhitungan *filtering* pada langkah *Pooling* (Santoso & Ariyanto, 2018).

Pada gambar 2.12 Merupakan proses lanjutan dari gambar 2.11 dimana satu persatu dari hasil *matrix* 2x2 terisi semua. *Filter* sudah diatur jika menemukan angka 4 *positive* maka merupakan sebuah informasi. Jadi setiap hasil perhitungan *matrix* yang bernilai 4 *positive* maka akan membentuk sebuah gambar seperti pada gambar 2.12. Kemudian untuk *matrix* di bawahnya menggunakan *rule* (-+ -+) dan mengisi setiap *matrix* 2x2 dengan aturan yang sama jika menemukan hasil 4 *positive* maka akan keluar hasil berupa informasi. Lanjutan proses hingga akhir dapat dilihat pada gambar 2.13.(Coskun et al., 2017) (Saha, 2018).



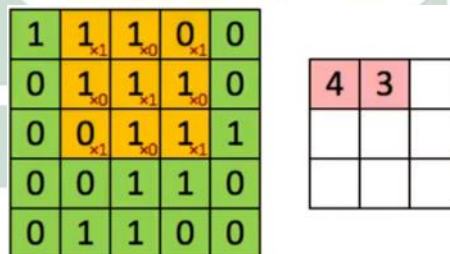
Pada gambar 2.13 merupakan keseluruhan dari proses perhitungan dengan *filter* 2x2 dengan rule (+-+-, -+-+) dan menghasilkan angka. Pada tahap ini merupakan proses *convolutional layer* proses *pooling layer* dimana setiap *kernel matrix* terisi dari hasil *filtering* dan *rule* yang sudah ada. Hasil dari angka tersebut menghasilkan sebuah pola baru. Langkah selanjutnya dapat dilihat pada Gambar 2.14 berikut (Coskun et al., 2017).



Gambar 2.15 *Convolutional Layer Step 1* (Eka Putra, 2016).

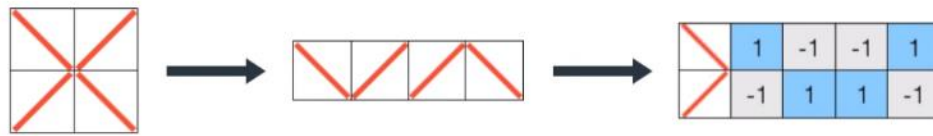
Pada gambar 2.15 Menjelaskan proses dari *Convolutional Layer* proses ini melakukan perhitungan dari *binary citra digital*, disetiap *citra digital* pasti terdapat angka berupa *binary*. Proses perhitungan yang dilakukan berupa pengaplikasian *kernel* (bagian berwarna *orange*) dimana mempunyai pola *matrix* perhitungan 3×3 . Pada bagian yang berwarna hijau, merupakan bagian *binary* yang akan dilakukan proses perhitungan *convolutional* (Eka Putra, 2016).

Jika setiap *kernel* sudah terhitung dengan model yang berlaku seperti pada gambar 2.15 maka akan diaplikasikan pada *matrix* 3 x 3 (Bagian Berwarna *pink*) sebagai proses *Pooling*, perhitungan *kernel* terus berlanjut seperti pada gambar berikut.



Gambar 2.16 *Convolutional Layer Step 2* (Eka Putra, 2016).

Gambar 2.16 merupakan proses lanjutan dari perhitungan *kernel* dari *binary* yang tersedia, jika hasil perhitungan *kernel* yang kedua sudah terhitung, maka masuk ke dalam *matrix Pooling*, proses perhitungan akan berlanjut hingga semua *binary* dan *matrix pooling* terisi penuh (O'Shea & Nash, 2015) seperti pada gambar berikut.

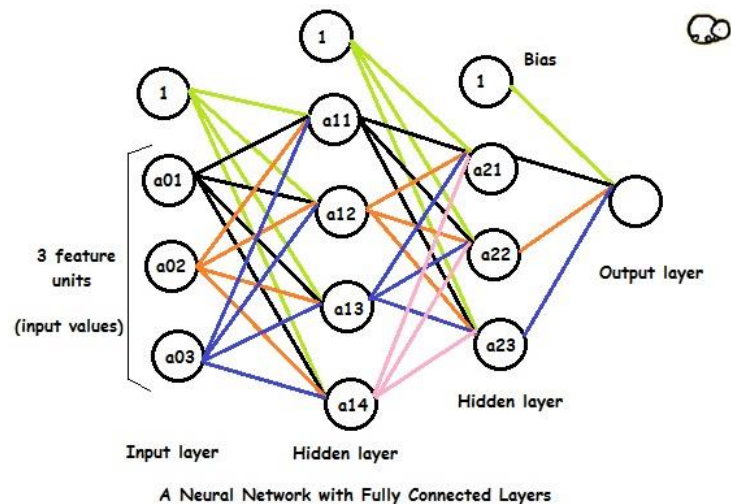


Gambar 2.20 Proses Flatten (Coskun et al., 2017) (Saha, 2018).

Pada gambar 2.20 merupakan proses lanjutan setelah melakukan proses *flatten* proses tersebut dilakukan untuk memvalidasi dari hasil *output*. Dengan cara mencocokkan pola yang sudah dihasilkan sesuai dengan hasil *flatten*. Jika bentuk pola sama dengan hasil *output flatten* maka menghasilkan nilai (+) jika tidak sama maka menghasilkan nilai (-). Aturan ini berbasis pada data *training* yang dilakukan sebelum dilakukannya proses pengujian sistem penggambarannya sesuai pada gambar 2.20 (Saha, 2018) (Coskun et al., 2017).

5. Fully Connected Layer

Fully connected layer merupakan proses lanjutan dari hasil *Flatten*, yaitu transformasi data menjadi bentuk *linear*, kemudian menghubungkan antar neuron menggunakan *backpropagation* dalam melakukan *classification* (Eka Putra, 2016).



Gambar 2.21 *Model Fully Connected 1 Layer* (Santoso & Ariyanto, 2018).

Pada gambar 2.21 merupakan konsep dari model *fully connected layer* dimana menggabungkan parameter yang sudah ada menjadi prose *deep learning* yang terdiri dari *input layer*, *fully connected layer (filter)*, dan *output layer*. Jika

The diagram shows a sequence of matrix operations. It starts with a 3x3 matrix:

$$\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}$$

This matrix is transformed into a 4x4 matrix:

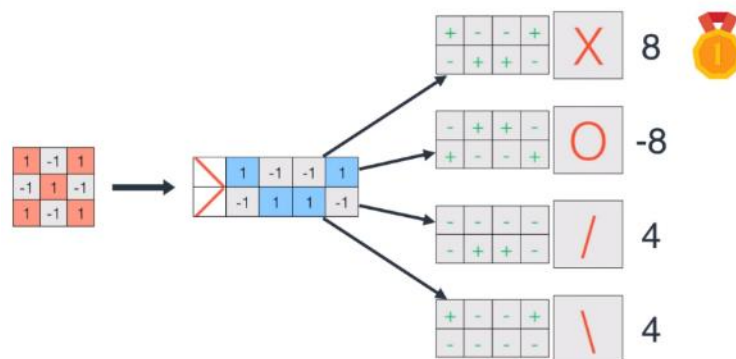
$$\begin{bmatrix} \text{X} & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

The 4x4 matrix is then transformed into four 4x4 matrices, each with a different operation (X, O, /, \) applied to the bottom-right 3x3 submatrix:

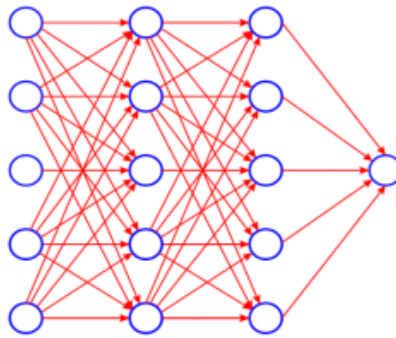
- Operation X: $\begin{bmatrix} + & - & + & + \\ - & + & + & - \\ + & - & - & + \\ + & - & - & - \end{bmatrix}$
- Operation O: $\begin{bmatrix} - & + & + & - \\ + & - & - & + \\ - & - & - & - \\ + & - & - & - \end{bmatrix}$
- Operation /: $\begin{bmatrix} - & - & - & - \\ - & + & + & - \\ - & - & - & - \\ + & - & - & - \end{bmatrix}$
- Operation \: $\begin{bmatrix} + & - & - & + \\ - & - & - & - \\ - & - & - & - \\ + & - & - & - \end{bmatrix}$

Below the 3x3 matrix, there are four green +1 symbols arranged in a 2x2 grid.

Pada gambar 2.22 merupakan gambaran dari model *fully connected layer* dari hasil proses *flatten* yang merubah dari model *kernel matrix* 2D menjadi 1D dan menghasilkan sebuah parameter baru dapat dihitung dan mengeluarkan hasil angka sesuai dengan prediksi gambar. Perhitungan tersebut sesuai dengan *rule* dari hasil *output flatten*. Hasil perhitungan dapat dilihat pada gambar 2.23 berikut.

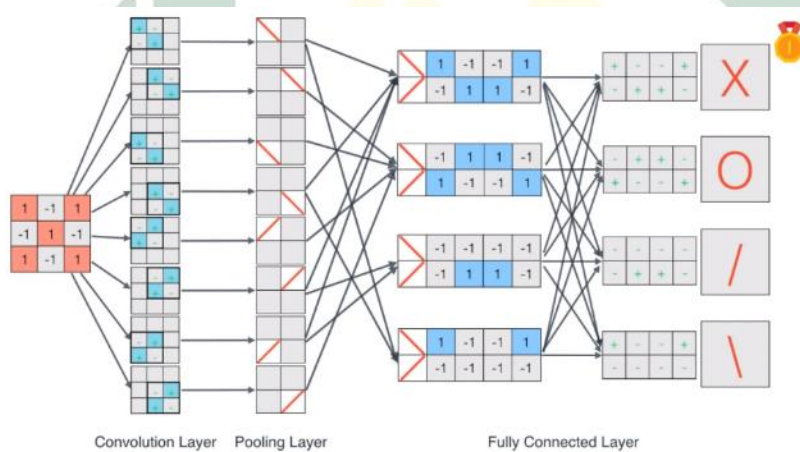


Pada gambar 2.23 Merupakan hasil dari perhitungan dengan *rule* yang ada. Jumlah angka yang paling besar merupakan angka yang terbaik dan sama dengan prediksi gambar (Saha, 2018). Dari gambar 2.22 hasil yang tertinggi adalah pola X. Proses tersebut merupakan *fully connected layer* pada langkah pertama, untuk langkah kedua yaitu dari proses perhitungan awal hingga akhir. Konsep tersebut dapat dilihat pada gambar 2.24 berikut.



Gambar 2.24 Model Fully Connected 2 Layer (Santoso & Ariyanto, 2018).

Pada gambar 2.24 merupakan proses *Fully Connected Layer*, konsep ini akan diaplikasikan dengan model algoritma *convolutional neural network* dari awal hingga akhir yang. Penggambaran model dapat dilihat pada gambar 2.25 sebagai berikut.



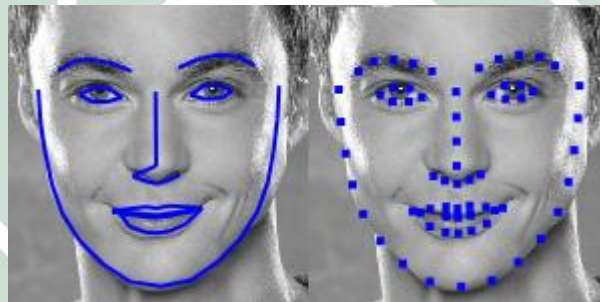
Gambar 2.25 Penerapan Fully Connected Layer Gabungan (Saha, 2018).

Pada gambar 2.25 merupakan proses keseluruhan dari penerapan *fully connected layer* dari awal hingga akhir. Pada gambar 2.25 dapat diambil kesimpulan bahwa dari hasil input berupa *matrix* 3x3 dan dengan dilakukan proses perhitungan dapat menghasilkan sebuah prediksi pola X yang paling tepat.

2.2.9. Face Recognition

Face recognition adalah salah satu teknologi yang terangkum dalam sistem biometrik yang telah banyak diaplikasikan dalam proses autentikasi maupun identifikasi, selain pengenalan retina mata, pengenalan sidik jari dan iris mata.

Dalam ruang lingkup aplikasinya sendiri pengenalan citra wajah dalam proses pengambilan gambarnya dibantu perangkat lain yaitu kamera untuk menangkap citra wajah seseorang kemudian dibandingkan dengan wajah yang sebelumnya (*Data Training*) yang telah disimpan di dalam database sistem (Angeline et al., 2019).



Gambar 2.26 *Face Landmark.*

Start to GUI

2.2.10. Face Expression

Ekspresi wajah merupakan hasil pendeteksian dari analisis emosi pada manusia, Tujuan dari adanya *face expression* untuk mengidentifikasi emosi dari manusia untuk menganalisis ekspresi yang sedang dirasakan, hal ini dibutuhkan oleh para konselor (Santhoshkumar & Kalaiselvi Geetha, 2019) (Phan-Xuan et al., 2019). Pakar beranggapan bahwa *face expression* mengungkapkan 2 hal tentang emosi pada manusia, yaitu yang pertama bahwa emosi merupakan reaksi terhadap masalah yang sudah terjadi, sebagai acuan penting untuk kesejahteraan, dan kedua, bahwa munculnya emosi pada manusia seringkali muncul dengan sangat cepat, yang tidak dapat disadari proses nya dalam lingkup pemikiran kita (Ekman, 2011:37) (Yang et al., 2020).

Hasil penelitian mengatakan mengenai ekspresi wajah pada manusia, dapat membuat para konselor merasa terbantu dalam mengatasi permasalahan yang sedang dihadapi oleh seseorang, terbukti jumlah dari 34 konseling yang melakukan proses bimbingan konseling, dapat dikatakan secara signifikan sempurna (100%) dan memberikan argumentasi bahwa konselor dengan ekspresi wajah dapat memberi informasi yang terkandung di dalamnya (Santhoshkumar & Kalaiselvi Geetha, 2019).

2.2.11. Model *Waterfall*

Metode *waterfall* merupakan sebuah model pengembangan *software* secara berurutan(*classic life cycle*). Pengembangan model jenis ini merepresentasikan pengembangan secara terstruktur dan juga saling berurutan secara teratur pada perancangan sistem yang akan dibangun. Proses awal yaitu dengan proses *requirement, specification, design, implementation, testing, maintainance* (Pressman, 2012).

system ini yaitu dari input kendaraan saat berada didekat *barrier gate system*. Dari penggunaan sensor maka *system* dapat bekerja semestinya.

2.2.14. Web Application

Aplikasi *web* merupakan perangkat lunak yang dikembangkan dengan komputer dan dikodekan dalam berbagai bahasa pemrograman yang mendukung perangkat lunak berbasis web seperti *HTML*, *CSS*, *JavaScript*, *Ruby*, *Python*, *Php*, *Java* serta bahasa pemrograman lainnya. Bahasa Pemrograman *HTML* (*Hypertext Markup Language*) merupakan kode *markup* standar yang sering digunakan untuk membangun aplikasi *web*. Banyak model dari pengembangan aplikasi *web* selalu ada struktur *HTML* di dalamnya. Oleh karena itu *HTML* adalah sebuah kode *markup* pemrograman yang wajib untuk dikuasai. Keunggulan *Web Application* berpotensi untuk berkolaborasi dengan segala pengembangan bahasa pemrograman (Lampropoulos et al., 2020).

Kemudian bahasa pemrograman lain yaitu *Java Script* yang fungsinya untuk mengolah halaman *web* untuk menangani aksi-aksi dari *user*, dan animasi yang dijalankan di sisi mesin *client*. *Javascript* juga salah satu bahasa pemrograman aplikasi *web* yang harus dikuasai. Sedangkan *CSS* atau (*Cascading Style Sheet*) merupakan kode untuk mendesain tampilan aplikasi *web* (*Style Sheet Language*) yang memiliki fungsi untuk mengatur tata letak dari halaman *web*, pewarnaan serta memperindah halaman *web*. *CSS* juga wajib dikuasai untuk pengembangan terutama pada *focus* untuk mendesain aplikasi *web*. Aplikasi *web* biasanya dibuat menggunakan kombinasi bahasa pemrograman yang dirancang untuk digunakan pada internet. Pada beberapa kasus, aplikasi *web* membutuhkan lebih sedikit *RAM* untuk dijalankan. Seperti sekarang ini, banyak *browser* permainan dibuat untuk dijalankan sebagai aplikasi *web*.

2.2.15. JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web yang dinamis dan interaktif. Inilah yang memberi gambaran dinamis pada halaman secara interaktif dan animasi yang dapat menarik perhatian pengunjung website. Ini adalah bahasa pemrograman web di sisi klien, yang berarti kode tersebut dijalankan di browser web pengguna. Dengan munculnya teknologi Node.js dan lainnya, ini juga dapat digunakan sebagai bahasa sisi server.

Berdasarkan kajian dalam ilmu Al – Qur'an dan pendapat Ustadzah Endang Rahayu, S. Pd. I selaku kepala TPQ Amanatul Islam serta sebagai Ustadzah pembimbing baca Al-Qur'an yang berlokasi di simowau indah Blok F Sepanjang-Taman-Sidoarjo beliau berpendapat mengenai tuntunan dari Firman Allah SWT yang harus kita teladani dari pemanfaat teknologi, hubungan sesama manusia, dan keamanan yang terdapat pada penelitian ini, dapat dilihat dari Firman Allah SWT Surah Al - Hujurât ayat 13:

Terjemah Kemenag 2002

Ayat ini menggambarkan antara hubungan sesama manusia dari berbeda suku, ras, budaya harus saling terjaga karena Allah SWT menciptakan kita sama derajatnya, jika dikaitkan dengan layanan *Human Resource* mempunyai integrasi, karena adanya system *Human Resource* dapat meningkatkan hubungan *social* antara manusia kemudian pada Surah Al – Anbiya (21:80).

Terjemah Kemenag 2002

Yang dimaksud adalah Allah Telah memberi cara kepada Nabi Dawud As yaitu membuat baju perisai yang berbahan besi, ini merupakan teknologi zaman dahulu di bidang peperangan, hal ini dapat di integrasikan bahwa, pemanfaatan teknologi, semata mata adalah anugerah Allah SWT maka harus digunakan sesuai tuntunan.

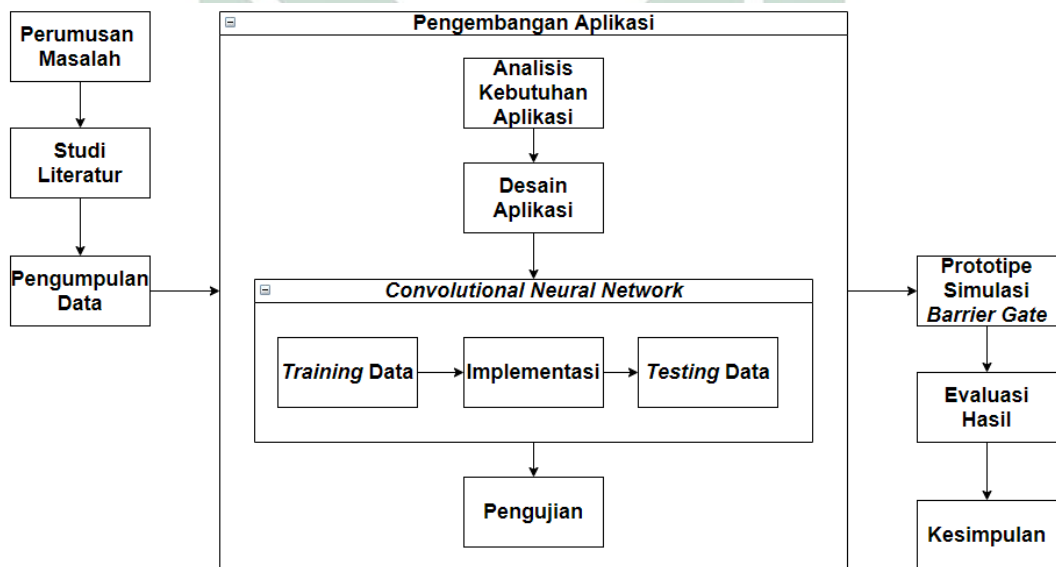
BAB III

METODOLOGI PENELITIAN

3.1. Desain Penelitian

Desain penelitian yang digambarkan dalam penelitian ini digambarkan dalam bentuk diagram alur, diagram ini digunakan untuk menggambarkan alur penelitian dengan alasan lebih mudah dalam hal penyampaian informasi. Diagram alur juga merupakan diagram yang mudah digunakan untuk menyampaikan konsep rancangan. Penelitian *face recognition* yang diaplikasikan dalam bentuk perangkat *web* dalam sistem autentikasi biometrik berbasis *face recognition* untuk simulasi *barrier gate* dibangun dengan diagram alur model *waterfall*. Model *waterfall* digunakan dalam penelitian ini untuk melakukan proses pengembangan aplikasi. Setiap proses memiliki spesifikasinya sendiri sehingga sistem dapat dikembangkan sesuai dengan apa yang dikehendaki (tepat sasaran).

Di dalam proses pengembangan aplikasi terdapat sebuah algoritma yang digunakan untuk melakukan proses klasifikasi yaitu menggunakan *convolutional neural network*. Berikut adalah digram alur dari desain penelitian yang dideskripsikan pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian.

Tabel 3. 2 Pengujian Math Static.

Tabel 3.2 menggambarkan sebuah pengujian dari *match statistic* dimana melakukan pengujian dengan cara menguji tingkat kecepatan sistem dalam menangkap sebuah citra kemudian menemukan *matching value* yang menggambarkan angka yang menunjukkan kecocokan angka dari *data training* yang merupakan sebuah parameter awal.

<i>Time</i>	<i>Face Identification</i>
<i>Minimum</i>	
<i>Maximum</i>	
<i>Average</i>	

42

Pengujian lain merupakan model pengujian dari ROC (*Receiver Operating Characteristic*) dimana didalamnya terdapat parameter TPR (*True Positive Rate*), FPR (*False Positive Rate*), FNR (*False Negative Rate*), TNR (*True Negative Rate*) model penggambaran tabel sebagai berikut.

Tabel 3. 4 Konsep ROC (Receive Operating Characteristic)

	<i>Condition Positive</i>	<i>Condition Negative</i>
<i>Predicted condition positive</i>	<i>True Positive Rate (TPR)</i>	<i>False Positive Rate (FPR)</i>
<i>Predicted condition negative</i>	<i>False Negative Rate (FNR)</i>	<i>True Negative Rate (TNR)</i>

Pengujian ROC fokus dalam pengujian pendeteksian secara akurat sesuai dengan gambar aslinya dan mendeteksi tingkat kesalahan yang dilakukan sistem dalam mengenali sebuah citra inputan berupa gambar maupun secara *real-time* yang dilakukan saat proses pengujian. Persamaan dari TPR, FPR, FNR, TNR untuk melakukan pengujian terhadap citra testing sebagai berikut:

$$TPR = \frac{\sum True\ positive}{\sum Condition\ positive} * 100 \quad (6)$$

$$FPR = \frac{\sum False\ positive}{\sum Condition\ negative} * 100 \quad (7)$$

$$FNR = \frac{\sum False\ negative}{\sum Condition\ positive} * 100 \quad (8)$$

$$TNR = \frac{\sum True\ negative}{\sum Condition\ negative} * 100 \quad (9)$$

Dari perhitungan persamaan tersebut dapat dilakukan pengujian model ROC dengan gambaran tabel sebagai berikut:

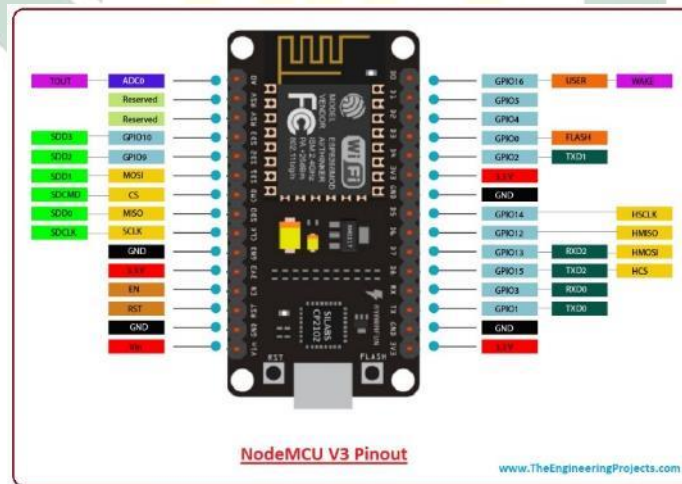
Tabel 3. 5 Tabel Pengujian ROC

Percobaan	Face Identification				Face Expression			
	TPR	FPR	FNR	TNR	TPR	FPR	FNR	TNR
Rata-rata								

Dari tabel 3.4 – 3.5 dapat menggambarkan tingkat kemampuan kecepatan sistem dalam mengidentifikasi citra dan mengetahui TPR, FPR, FNR, TNR dari *face recognition* dan *face expression* dapat mengetahui tingkat kemampuan sistem dalam mengidentifikasi secara tepat.

3.1.10. Simulasi *Barrier Gate System*

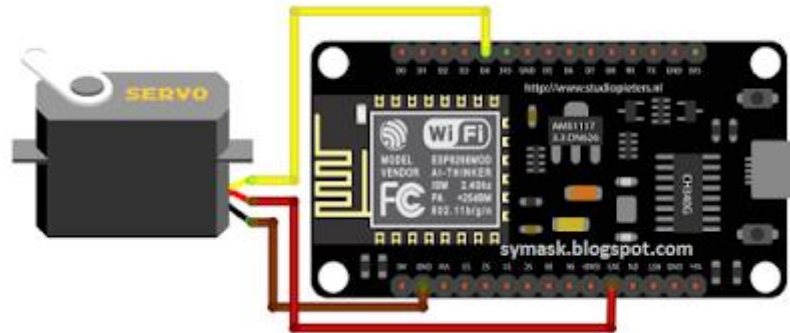
Simulasi *barrier gate* merupakan bentuk implementasi dari hasil *face recognition* dan *face expression*. Proses simasi ini menggunakan *microcontroller* Node MCU, *servo* dan beberapa rangkaian elektronika yang membentuk sebuah *barrier gate system* sebagai integrasi dari hasil *output face recognition* ke dalam *hardware*, berikut penggambaran model dari *microcontroller* NodeMCU.



Gambar 3.2 Pin NodeMCU (Cyntia Windiasari et al., 2019).

Pada gambar 3.2 menggambarkan mengenai pin pin yang terdapat pada *microcontroller* NodeMCU setiap pin mempunyai fungsi yang berbeda beda. Terdapat Vcc, Gnd dan pin lain. Dalam fungsi simulasi *barrier gate system* tidak semua *pin* digunakan, hanya membutuhkan beberapa *pin* saja yang akan

disambungkan dengan *servo*. Berikut penggambaran dari model *servo* yang dihubungkan dengan *servo*.



Gambar 3.3 *microcontroller* dan *servo* (Cyntia Windiasari et al., 2019).

Pada gambar 3.3 menggambarkan rangkaian untuk menghubungkan dari *microcontroller* ke *servo*. Servo mempunyai 3 inputan utama yang terdiri dari Vcc, Gnd, dan Data. Vcc merupakan input *positive* dari *microcontroller* dengan tegangan 3,3 Volt. Gnd untuk menyambungkan ke *ground* yang bermuatan listrik *negative*. Data merupakan sebuah inputan dari *servo* untuk menggerakkan berapa derajat kemiringan yang dikontrol oleh *microcontroller*.

3.1.11. Evaluasi Hasil

Tahap akhir dari metodologi penelitian yaitu evaluasi hasil dengan cara uji coba di lapangan menggunakan sistem yang sudah ada. Dalam melakukan pengujian sistem, perekaman data citra harus sudah terekam pada sistem, kemudian dilakukan proses pengidentifikasian tingkat kecepatan sistem dalam mengenali setiap citra wajah yang dengan melakukan uji performa sistem.

3.2. Waktu Dan Tempat Penelitian

dan tempat penelitian ini dilakukan di Universitas Islam Negeri Sunan Ampel Surabaya. Penelitian dimulai pada bulan September 2019, hingga sampai tahap publikasi yang akan datang pada bulan April 2020.

4.2. Analisis Kebutuhan

Berdasarkan penulisan latar belakang dan rumusan masalah yang diajukan terkait dengan Autentikasi biometrik berbasis *face recognition* menggunakan metode *convolutional neural network* untuk simulasi *barrier gate system*, menghasilkan analisis kebutuhan sistem berupa *software* dan *hardware* yang digunakan sebagai pendukung dari penelitian ini sebagai berikut:

Tabel 4. 1 Kebutuhan *Hardware*

No	Hardware	Minimal System Requirements
1	Laptop	Operating System Windows 7 Ultimate 64-bit or 32-bit
		Processor AMD A4-3305M APU with Radeon HD Graphics
		Frequency 1.9 Ghz
		RAM 4096 Mb
		HDD 500 GB
		Connectivity USB 2.0
2	Webcam	8 Mega Pixels Resolution Camera
		30 fps Maximum Frame Rate
		Stand Installation Type
		640 x 480 Pixels Resolution Video
		Connectivity USB 2.0
3	NodeMCU	Microcontroller Chip ESP8266 - 12E
		Input Voltage 7V – 12V
		Voltage 3.3V – 5V
		GPIO 13 PIN
		Flash Memory 4 Mb
		SRAM 64 Kb
		Clock Speed 80 Mhz
		Include Wifi IEEE 802.11 b/g/n
4	Servo	Operating Voltage 4.8V - 7.2V
		Operating Current 100Ma
		Operating Speed 0.13sec/60 Degree 6V 0.17/60 Degree 0.17
		Torque 13. Kg/Cm 4.8V

Pada tabel 4.1 merupakan spesifikasi minimum hardware yang digunakan dan sebagai gambaran dari kebutuhan sistem dalam pendukung penelitian ini. Kebutuhan spesifikasi pada tabel tersebut merupakan spesifikasi dasar dari

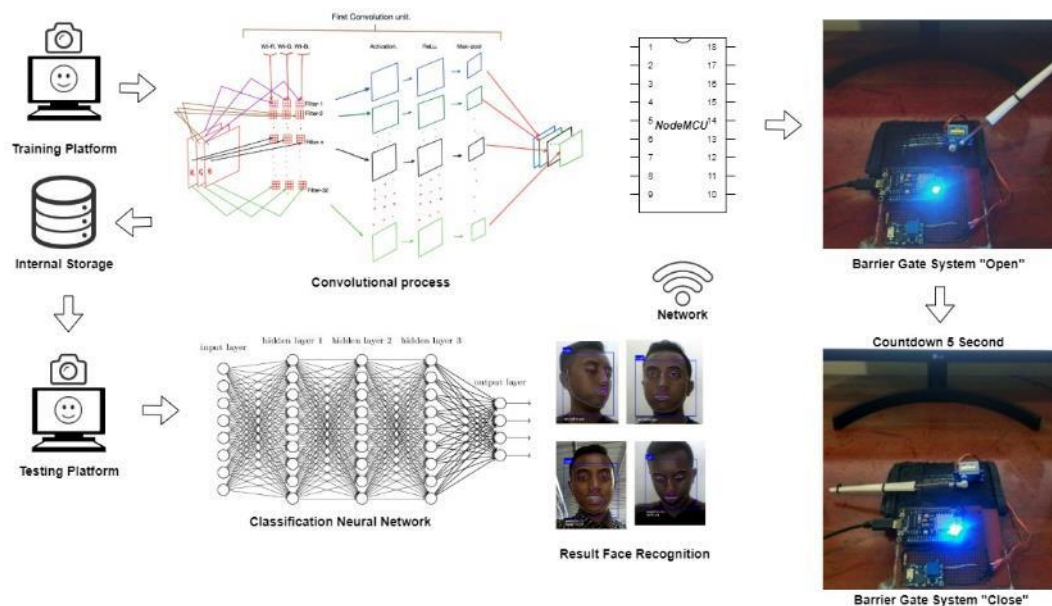
hardware yang dapat digunakan pada proses pengembangannya. Selain kebutuhan *hardware*, kebutuhan *software* yang digunakan yaitu sebagai berikut:

1. Visual Code Studio
2. Web Server Internal Visual Code Studio
3. Sublime
4. MySQL
5. Library Face Recognition
6. Arduino.IDE
7. Postman

Dengan beberapa detil jenis *hardware* dan *software* menjadi sebuah kebutuhan dalam pembangunan aplikasi yang mempunyai fungsi utama sebagai autentikasi keamanan menggunakan model biometrik berbasis *face recognition*, jadi tidak ada fasilitas menu, karena hanya difungsikan sebagai autentikasi dari data latih yang sudah dilakukan terlebih dahulu.

Saat proses autentikasi pada sistem, *user* harus menghadap ke kamera, kemudian sistem akan melakukan proses *scanning* citra inputan, menggunakan metode *convolutional neural network*, kemudian sistem akan secara otomatis membandingkan dengan *training data* yang sudah ada didalam *directory system*, jika *testing data* sesuai dengan *training data* maka *barrier gate system* akan membuka, jika *testing data* dan *training data* tidak sesuai maka *barrier gate system* tetap menutup. Ketika sudah didalam aplikasi, sistem akan secara otomatis mendeteksi dengan menampilkan titik wajah berdasarkan wajah yang sedang diuji, dan memunculkan nama serta ekspresi yang sedang dilakukan oleh *user*. Penampilan titik wajah dan identifikasi *user* dihasilkan dari metode Convolutional Neural Network (CNN) untuk menentukan setiap perhitungan *binary pixels*.

Proses pengiriman hasil output aplikasi *face recognition* akan dikirimkan melalui jaringan *wifi* dengan memanfaatkan fitur dari NodeMCU berdasarkan modul ESP8266 yang digunakan untuk keperluan *web server*. *Webserver* pada mikrokontroler merupakan sebuah pemanfaatan dari *Internet of Things*, dan merupakan konsep menghubungkan satu perangkat dengan perangkat lain menggunakan Internet atau jaringan lokal.



Gambar 4. 3 Desain Arsitektur *System*.

Desain arsitektur sistem merupakan desain arsitektur sistem yang sedang digunakan, proses awal yaitu melakukan *training data* kemudian dilakukannya proses *convolutional neural network* yang terdiri dari *convolution*, *filtering*, *pooling*, *flatten*, kemudian disimpan di penyimpanan internal. Proses autentikasi *face recognition* menggunakan *classification neural network* yang disamakan dengan *training data* sebelumnya. Jika wajah terdeteksi, maka akan mengirimkan sinyal untuk menggerakkan *barrier gate system*, kemudian selama 5 detik otomatis akan menutup kembali.

4.3.3. Desain Arsitektur *Software*

Desain arsitektur software merupakan gambaran mengenai aplikasi yang dimodelkan berdasarkan *software – software* yang digunakan. *Software* tersebut merupakan penunjang dalam aplikasi ini untuk mendapatkan hasil yang akurat, penggambaran diagram sebagai berikut:



Gambar 4. 5 Desain Arsitektur *Face Recognition*.

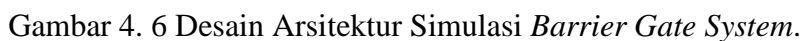
Pada gambar 4.5 terjadi 2 proses dalam lingkup *face recognition*, yaitu *training data* dan *testing data*. Proses *train data* merupakan proses awal dalam melakukan pembelajaran terhadap sistem dalam mengenali sebuah citra inputan *digital*. Proses tersebut dilakukan dengan data dari setiap sampel foto dengan teknik pengambilan seperti yang sudah tertulis. Setiap individu mempunyai 10 foto yang berbeda dan akan dilakukan proses *convolution*, yaitu perpindahan dan proses identifikasi *pixels* dari sebuah citra *input*. Masing masing gambar terlihat mempunyai *kernel matrix*, pada *convolution* pertama mempunyai *pixels* 460x480 px kemudian dilakukan proses *filtering* menggunakan kernel matrix 12x12 (Yang et al., 2020).

Setelah selesai pada proses *convolution* pertama, maka melakukan proses *convolution* yang kedua hingga proses akhir yaitu *flattening*. Pada setiap *feature map* dan proses *convolution* melakukan proses *filtering binary matrix* dengan mengidentifikasi warna warna dasar yaitu *red, green, blue* (RGB). Sehingga tidak perlu dilakukan proses perubahan ke warna *grey*. Proses filter tersebut dilakukan secara bersamaan. *Kernel channel* dibedakan menjadi 3 model *filter* yang sudah

5. Desain Arsitektur Prototype Barrier Gate System

Desain arsitektur berikut merupakan visualisasi dari proses kerja dari simulasi *prototype barrier gate system*. Model arsitektur simulasi ini digambarkan dengan simbol – simbol dari komponen elektronika dengan tujuan memvisualisasikan informasi dari model arsitektur yang dirancang pada penelitian ini. Model diagram ini dapat dilihat pada gambar berikut.

Desain arsitektur berikut merupakan visualisasi dari proses kerja dari simulasi *prototype barrier gate system*. Model arsitektur simulasi ini digambarkan dengan simbol – simbol dari komponen elektronika dengan tujuan memudahkan informasi dari model arsitektur yang dirancang pada penelitian ini. Model diagram dapat dilihat pada gambar berikut.



53

Hasil dari proses output *Face Recognition* dari *Personal Computer*, akan dikirimkan ke *network* yang akan diterima oleh NodeMCU ESP8266 dan menerimanya sebagai *data* untuk menggerakkan micro servo S90. NodeMCU ESP8266 diprogram dalam menerima *response* dari *web*, *response* tersebut berupa variabel *True* dan *False*. Jika *response* dari *web* sama dengan *true* maka program yang terdapat pada NodeMCU ESP8266 akan menjalankan *function* buka dan sebaliknya jika *response* dari *web* maka akan menjalankan *function* tutup.

4.3.6. UML Use Case Diagram

54

dapat melakukan proses *face detection* saja, dan *trainer* dapat melakukan penambahan data wajah baru beserta label nama sesuai dengan aturan pengambilan data.

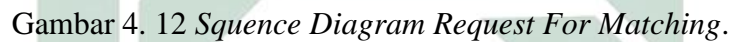
Gambar 4. 7 Use Case Diagram.

4.3.7. UML Activity Diagram

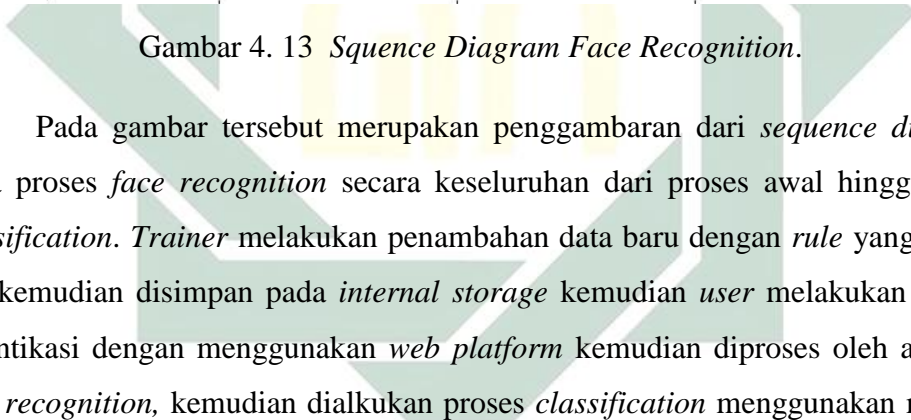
Interaksi antara *object* yang berperan dalam aplikasi ini digambarkan dalam model UML *sequence diagram*. Pada diagram ini terdapat 4 model utama yang digambarkan dengan model *sequence diagram* ini yaitu, *add face data*, *request matching*, dan *face recognition* yang digambarkan sebagai berikut.



Jika proses *add face data* sudah dilakukan maka *user* akan melakukan proses *request matching* yaitu proses autentikasi. Pada proses ini mempunyai kesamaan dengan diagram sebelumnya yaitu mempunyai 4 *object* dan 1 *actor* proses interaksi yang terjadi pada proses tersebut digambarkan dalam model *sequence diagram* sebagai berikut:



Pada proses berikutnya yaitu menggambarkan bagaimana proses *face recognition* dalam melakukan *classification* data menggunakan metode CNN. Diagram ini merupakan detil dari keseluruhan proses yang terjadi antar *object* yang digambarkan pada *sequence diagram*, perbedaan dari model diagram sebelumnya yaitu terletak pada aktor nya, pada diagram ini terlihat mempunyai 2 aktor, yaitu *user* dan *trainer* yang digambarkan sebagai berikut:



Gambar 4. 13 *Sequence Diagram Face Recognition.*

Pada gambar tersebut merupakan penggambaran dari *sequence diagram* pada proses *face recognition* secara keseluruhan dari proses awal hingga hasil *classification*. *Trainer* melakukan penambahan data baru dengan *rule* yang sudah ada kemudian disimpan pada *internal storage* kemudian *user* melakukan proses autentikasi dengan menggunakan *web platform* kemudian diproses oleh aplikasi *face recognition*, kemudian dilakukan proses *classification* menggunakan metode CNN dari proses *Convolutional layer* hingga proses *fully connected layer* yang akan mengeluarkan hasil dengan cara menampilkan hasil output pada aplikasi.

4.3.9. UML Class Diagram

Class Diagram merupakan konsep diagram yang bersifat terstruktur yang sering digunakan untuk menggambarkan rancangan *class* yang digunakan dalam aplikasi. Model *class diagram* digambarkan dalam model sebagai berikut:

2. Authentication

Class tersebut hanya mempunyai tipe data *string* yang bersifat *public*, kemudian diolah menggunakan *method* POST yang akan mengirimkan ke alamat *url* tertentu.

3. *TrainFunction*

Pada *class* tersebut ada beberapa jenis variabel yang digunakan yaitu Xtrain, Ytrain, dan trainID dengan model tipe data *float* pada proses tersebut pada trainID merupakan proses perhitungan pada CNN dengan menggunakan angka yang bersifat *decimal*.

4. Recognition Function

Class tersebut mempunyai beberapa model yaitu *cariID* dengan tipe data *string*, *angk* dengan tipe data *Integer*, *checkNameID*, *checkXtrain*, dan *checkExpressionID* yang membutuhkan masukan dari *class* lain kemudian mempunyai relasi *One to Many*.

5. Expression Function

Class tersebut mempunyai fungsi yang hampir sama dengan *recognition function* dan mempunyai *expressionID* dengan tipe data *string* kemudian beberapa variabel data ekspresi yaitu *angry*, *disgusted*, *fearful*, *happy*, *neutral*, *sad*, *surprised* yang masing masing mempunyai tipe data *string*.

6. Face API

Class tersebut berisikan banyak proses dari berbagai kelas yaitu *checkNameID()*, *checkExspressionID()*, *checkLabelID()* kemudian mendapatkan hasil *output* yang akan dikirimkan ke Xml Http Request yang akan direspon oleh *hardware*.

7. *XmlHttpRequests*

Class tersebut mempunyai beberapa variabel yaitu *varXhr* dengan model tipe data *variable character* kemudian *getResultRecognition()* yaitu mengambil hasil dari proses *recognition* dan *getResultExpression()* mengambil hasil dari proses *expression* yang kemudian dikirimkan ke *url* menggunakan method **POST**.

Inti proses pada *pseudocode* pada step 6 yaitu proses feture learning, yang didalamnya terdapat 4 proses utama yaitu *convolution*, *pooling*, *max pooling*, dan *classification using neural network* semua proses tersebut terjadi secara otomatis yang dilakukan oleh sistem yang bekerja sesuai dengan *script* yang sudah dibuat.

4.3.10.3. Xml Http Request

Pada *function* XML merupakan proses perpindahan variable dari program *javascript* yang dikirim ke *web platform* dengan menggunakan Http Request yang tersedia oleh NodeMcu, berikut penjelasannya digambarkan dalam bentuk *pseudocode* sebagai berikut:

```

1. GET new [valueOpen] = FOR XMLHttpRequest();
2. IF [DATA] faceRecognition == true, Then DO Step 3 Else Do Step 8
3. [Value].open("POST", "http://192.168.43.254/servoA?value=50");
4. [Value].send(Signal FOR open);
5. WHILE open, DO countdown = setTimeout(function(In 5 Second)
6. if (seconds <= 0)Else DO step 8
7. Else Do Step 6
8. GET new [valueClose] = FOR XMLHttpRequest(Close);
9. [valueClose].open("POST", "IP Address Close Signal Base Router");
10. [valueClose].send(Signal FOR Close);

```

Proses inti dari *pseudocode* response tersebut pada langkah 2. Jika hasil *output face recognition* sama dengan *true*, maka variable akan dikirim ke url sebagai berikut <http://192.168.43.254/servoA?value=50> dan mengirim sinyal buka ke NodeMCU ESP8266 jika hasil *output face recognition* sama dengan *false* atau tidak terdeteksi wajah, maka akan menghitung mundur selama 5 detik dan mengirimkan sinyal menutup.

4.4. Prototype Simulasi Barrier Gate System

Prototype Simulasi Barrier Gate System akan disimulasikan menggunakan komponen elektronika yang dirangkai sesuai dengan ilmu elektronika dasar, dalam proses pembuatan skematik rangkaian menggunakan *software* Fritzing, untuk mendapatkan model skematik setiap komponen yang dibutuhkan, perlu dilakukan proses pencarian *library* secara *online* maupun *offline* sesuai dengan nama komponen.

Komponen dasar yang digunakan dalam proses prototype simulasi *barrier gate system* sebagai berikut:

- Komponen elektronika tersebut kemudian dirangkai seperti gambar yang divisualisasikan pada gambar berikut.



67

tersebut merupakan hasil dari proses skema rangka

out GPIO2 TXD1 / D4 merupakan *input* gelombang digital, jika *output face recognition* bernilai *true*, maka *pinout* GPIO2 TXD1 / D4 akan bernilai *High* (On), saat proses tersebut *pinout* GPIO2 TXD1 / D4 tegangan menurun menjadi 0,2 Volt DC. Jika *output face recognition* bernilai *false* maka *Micro Servo Motor SG90* bernilai *Low* (Off), saat proses tersebut *pinout* GPIO2 TXD1 / D4 tegangannya naik menjadi 0,5 Volt DC.

Untuk melakukan proses *convolutional layer* dan mengolah data dari hasil *binary matrix* awal dapat menggunakan model *Filter* (3x3) dengan matrix [1, 0, 1, 0, 1, 0, 1, 0, 1] sebagai berikut (Eka Putra, Wayan Suartika 2016)

1	0	1
0	1	0
1	0	1

Model *filter* (3x3) tersebut dapat dijadikan untuk melakukan proses perhitungan *matrix* baru dan menghasilkan *Convolved feature matrix* dengan perhitungan menggunakan persamaan sebagai berikut:

$$\sum F P n = ((m1 \ c1) \times f1) +((mn \ cn) \times fn)$$

- a) *Position 1* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- b) *Position 2* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (1x1) = 0$
- c) *Position 3* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- d) *Position 4* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- e) *Position 5* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- f) *Position 6* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- g) *Position 7* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) = 0$
- h) *Position 8* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x0) + (1x1) = 1$
- i) *Position 9* : $(0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (1x0) + (0x1) + (1x0) + (1x1) = 1$
- j) *Position 10* : $(0x1) + (0x0) + (0x1) + (0x0) + (1x1) + (1x0) + (1x1) + (1x0) + (1x1) = 3$

$$FMn = \left(\sum FPrn\right) + \left(\sum FPgn\right) + \left(\sum FPbn\right) + b \quad (11)$$

$$\sum FPn = ((m1\ c1) \times f1) + ((mn\ cn) \times fn) \quad (12)$$

$$\begin{aligned} FMn & : \text{Feature Map } x \\ \sum FP & : \text{Total Feature Map} \\ \sum FP_{rn} & : \text{Total Feature Map Red } x \\ \sum FP_{gn} & : \text{Total Feature Map Green } x \\ \sum FP_{bn} & : \text{Total Feature Map Blue } x \end{aligned}$$

Dari persamaan berikut dapat dilakukan proses perhitungan *red, green, blue* berdasarkan hasil *convolved feature matrix* dengan *filter* RGB masing masing sebagai berikut:

-1	-1	1
0	1	-1
0	1	1

1	0	0
1	-1	-1
1	0	-1

0	1	1
0	1	0
1	-1	1

Gambar 4. 22 *Filter Matrix* RGB (Coskun et al., 2017) (Saha, 2018).

Filter tersebut dijadikan sebagai acuan dalam menentukan hasil *convolved feature RGB matrix*. Proses perhitungan dalam mencari *convolved feature RGB matrix* sebagai berikut:

$$\text{PIR: } (0x(-1)) + (0x(-1)) + (0x1) + (0x0) + (0x1) + (0x(-1)) + (0x0) + (0x1) + (0x1) = 0$$

$$\text{PlG: } (0x1) + (0x0) + (0x0) + (0x1) + (0x(-1)) + (0x(-1)) + (0x1) + (0x0) + (0x(-1)) = 0$$

$$\text{PIB: } (0x0) + (0x1) + (0x1) + (0x0) + (0x1) + (0x0) + (0x1) + (0x(-1)) + (0x1) = 0$$

$$FM1 = (0) + (0) + (0) + 1 = 1$$

2	12	19	24	22	21	21
17	17	9	7	15	17	20
13	8	6	1	5	9	11
15	15	15	6	17	20	20
11	14	15	13	24	23	25
9	10	12	8	12	24	19
2	3	6	15	15	9	14
1	6	11	22	23	19	25
1	5	15	16	18	20	22
1	3	8	13	19	23	13
1	2	5	10	22	25	8

Dari hasil proses *max pooling* tersebut menjadi nilai paling tinggi, dari hasil *convolved feature RGB matrix* sebelumnya, hasil *max pooling matrix* tersebut akan dilanjutkan pada proses *flattening*.

Proses *flattening* merupakan proses perubahan dimensi pada matrix yang awal nya mempunyai *matrix* 2D diratakan menjadi model 1D *matrix* yang akan diteruskan pada proses *fully connected layer*, berikut merupakan proses *flattening*

2	12	19	24	22	21	21	2
17	17	9	7	15	17	20	12
13	8	6	1	5	9	11	19
15	15	15	6	17	20	20	24
11	14	15	13	24	23	25	22
9	10	12	8	12	24	19	21
2	3	6	15	15	9	14	21
1	6	11	22	23	19	25	17
1	5	15	16	18	20	22	17
1	3	8	13	19	23	13	...
1	2	5	10	22	25	8	8

[illegible]

4.5.5. Fully Connected Layer

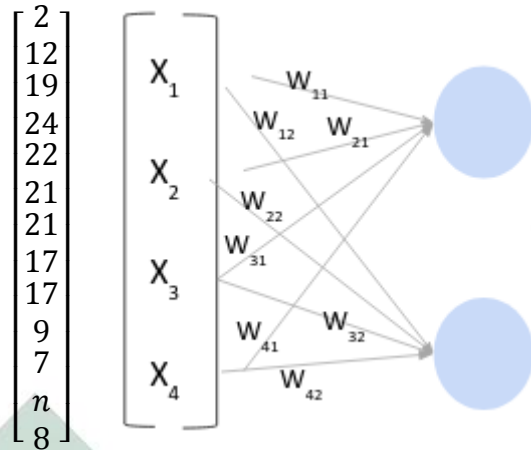
Dalam proses perhitungan yang ada pada konsep *neural network* mempunyai 2 bagian utama, yakni *forward propagation*, dan *backward propagation*. *Forward propagation* akan menghasilkan nilai y_n (*output*), sedangkan pada proses *backward propagation* akan melakukan update pada W_n (*weight*) sehingga pada prosesnya dapat menghasilkan hasil *output* yang optimal berikut proses perhitungan *fully connected layer* menggunakan konsep dasar *neural network*.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} \quad W = \begin{bmatrix} W_1 & W_7 \\ W_2 & W_8 \\ W_3 & W_9 \\ W_4 & W_{10} \\ W_6 & W_{11} \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$\begin{aligned} X_n &: \text{Input Data} \\ W_n &: \text{Weight} \\ b_1 &: \text{bias} \end{aligned}$$

78

2	12	19	24	22	21	21
17	17	9	7	15	17	20
13	8	6	1	5	9	11
15	15	15	6	17	20	20
11	14	15	13	24	23	25
9	10	12	8	12	24	19
2	3	6	15	15	9	14
1	6	11	22	23	19	25
1	5	15	16	18	20	22
1	3	8	13	19	23	13
1	2	5	10	22	25	8



Gambar 4. 26 *Input Value Pada Neural Network.*

Model persamaan diatas merupakan hasil dari proses *flattening* yang merubah *array matrix* menjadi 1D, dan dilakukan proses pemasukan pada tahap *fully connected layer*. Untuk tahap pertama akan menentukan dengan proses perhitungan *forward propagation* dengan model persamaan sebagai berikut:

$$y_j = \sigma \left(\sum_{i=1}^4 W_{ij} X_i + b_j \right) \quad (13)$$

y_j : Output
 σ : Sigma (hasil perkalian *input* dan *weight*)
 W_{ij} : Weight
 X_i : Input
 b_j : bias

Dari persamaan proses perhitungan *forward propagation* tersebut dapat diartikan dengan model persamaan berikut:

$$\begin{aligned} y_1 &= \sigma(w_{11}x_1 + w_{21}x_2 + w_{31}x_1 + w_{41}x_3 + b_1) \\ y_2 &= \sigma(w_{12}x_1 + w_{22}x_2 + w_{32}x_1 + w_{42}x_3 + b_2) \\ y_3 &= \sigma(w_{13}x_1 + w_{23}x_2 + w_{33}x_1 + w_{43}x_3 + b_3) \end{aligned} \quad (14)$$

$$\sum_j e^{a_j} = e^{a_j} + e^{a_j} + e^{a_j} + \dots n \quad (16)$$
$$\sum_j e^{a_j} : \text{Hasil penjumlahan dari } e^{a_i}$$
$$\sum_j e^{a_j} = 2.022.360.730.323,5$$
$$softmax(\sigma) = [0.01, 0.5, 0.6]$$
[illegible]

perhitungan berdasarkan hasil dari perhitungan sebelumnya diperoleh nilai y_n sebagai berikut:

$$[y_1 = 0.01, y_2 = 0.5, y_3 = 0.6]$$

Nilai tersebut merupakan nilai prediksi yang dihasilkan dari proses *neural network* pada tahap *fully connected layer* CNN yang sudah dilakukan aktivasi menggunakan *softmax function*. Dari hasil nilai y_n yang didapat, untuk jenis input x_n harus mempunyai model yang bernilai sebagai berikut:

$$[t_1 = 0 \ t_2 = 0.25 \ t_3 = 1]$$

t_n merupakan simbol dari nilai *output* yang seharusnya pada masing masing *neuron*. Dari hal tersebut dapat dilakukan proses perhitungan *error* untuk mengetahui hasil prediksi menggunakan persamaan *mean square error* sebagai berikut:

$$error = \frac{1}{3} \sum_{i=1}^3 (target_i - prediksi_i)^2 \quad (17)$$

Persamaan tersebut untuk melakukan proses perhitungan selisih dari nilai target dan prediksi, kemudian dilakukan pengkuadratan, dan melakukan rata rata dari ketiga variabel yang sudah disebutkan. Angka 3 didapat dari proses *neural network* pada proses *fully connected layer* yang sudah dilakukan perhitungan diatas. Sehingga proses perhitungannya sebagai berikut:

$$error = \frac{1}{3} \sum_{i=1}^3 (target_i - prediksi_i)^2$$

$$E = \frac{1}{3} \sum_{i=1}^3 (target_i - prediksi_i)^2$$

$$E = \frac{1}{3} ((0 - 0.01)^2 + (0.25 - 0.5)^2 + (1 - 0.6)^2)$$

$$E = 0.26$$

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w} \quad (18)$$

- α = Konstanta (0.5)
- $\frac{\partial E}{\partial b}$ = Turunan Parsial E terhadap w
- w = *weight*

$$b_{new} = b_{old} - \alpha \frac{\partial E}{\partial b} \quad (19)$$

α	= Konstanta (0.5)
$\frac{\partial E}{\partial b}$	= Turunan Parsial E terhadap w
b	= Bias


$$\frac{\partial E}{\partial w_{11}} = \frac{\partial E}{\partial y_{11}} \frac{\partial y_{11}}{\partial z_1} \frac{\partial z_1}{\partial w_{11}} \quad (20)$$

∂	: Turunan Parsial
E	: <i>Error</i>
w	: <i>Weight</i>
y_n	: Nilai Prediksi
z_n	: <i>Total</i> $\sum(w_n \cdot x_1 + \dots n + b)$
x_n	: Input <i>flattening matrix</i>

[illegible]

$$\frac{\partial E}{\partial w_{11}} = 0$$

$$\frac{\partial E}{\partial w_{12}} = 0.45$$

$$\frac{\partial E}{\partial w_{13}} = -1.0944$$

$$\begin{aligned}w_{new\ 11} &= w_{old} - \alpha \frac{\partial E}{\partial w} \\w_{new\ 11} &= 0.1 - (0.5) \cdot (0) \\w_{new\ 11} &= 0.1\end{aligned}$$

$$\begin{aligned}w_{new\ 12} &= w_{old} - \alpha \frac{\partial E}{\partial w} \\w_{new\ 12} &= 0.2 - (0.5) \cdot (0.45) \\w_{new\ 12} &= 0\end{aligned}$$

$$w_{new\ 13} = 0.6472$$

am proses *backpropagation* memiliki tujuan utama yaitu melakukan *update weight* dan *bias* yang akan diteruskan untuk melakukan perhitungan berikutnya. Jadi pada metode CNN di dalam proses *fully connected* dilakukan proses perhitungan *neural network* dengan menggunakan 2 konsep *forward propagation* dan *backward propagation*. *Forward propagation* menentukan nilai y_n serta menemukan *error* antara y_n dengan t_n , dan *backward propagation* untuk melakukan pembaharuan *weight* dan perhitungan yang sudah dilakukan merupakan keseluruhan dari metode CNN proses *convolutional layer* hingga *fully connected layer* untuk proses *classification*. Proses CNN yang terekam oleh aplikasi dapat dilihat pada gambar berikut.

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 26, 26, 32)	320
conv2d_8 (Conv2D)	(None, 24, 24, 64)	18496
conv2d_9 (Conv2D)	(None, 22, 22, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 11, 11, 128)	0
dropout_5 (Dropout)	(None, 11, 11, 128)	0
flatten_3 (Flatten)	(None, 15488)	0
dense_5 (Dense)	(None, 128)	1982592
dropout_6 (Dropout)	(None, 128)	0
dense_6 (Dense)	(None, 10)	1290
Total params: 2,076,554		
Trainable params: 2,076,554		
Non-trainable params: 0		

Gambar 4. 28 Hasil Perhitungan CNN Pada System.

apatkan jumlah total parameter sebanyak 2.076.554 yang akan proses *classification* pada *neural network*.

Hasil *recognition* berdasarkan model training data Berikut merupakan hasil *face recognition* dari *training data* dan dilakukan percobaan menggunakan model data yang n model data yang berbeda, berikut visualisasinya:





4. 2 Percobaan *Face Recognition*

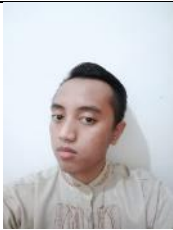









Training Data	Result	Toleransi	Keter
		Face 0,24 Neutral 0.99	Posisi peng menghadap k lurus dan de yang datar. Me yang sesuai d 0.24 & 0.99









4.5.6. Hasil *recognition* berdasarkan model training data

Berikut merupakan hasil *face recognition* dari *training data* yang sudah dilakukan dan dilakukan percobaan menggunakan model data yang sama dan dengan model data yang berbeda, berikut visualisasinya:

Tabel 4. 2 Percobaan *Face Recognition*

No	Training Data	Result	Toleransi	Keterangan
1			Face 0,24 Neutral 0.99	Posisi pengambilan foto menghadap kamera dengan lurus dan dengan ekspresi yang datar. Mendapatkan hasil yang sesuai dengan toleransi 0,24 & 0,99.
2			Face 0,26 Happy 0.99	Posisi pengambilan foto menghadap kamera dengan lurus dan dengan ekspresi tersenyum. Mendapatkan hasil yang sesuai dengan toleransi 0,30 & 0,99.

3			Face 0,24 Neutral 0,99	Posisi pengambilan foto hadap sedikit ke kanan dan dengan ekspresi datar. Mendapatkan hasil yang sesuai dengan toleransi 0,24 & 0,99
4			Face 0,34 Happy 0,99	Posisi pengambilan foto hadap sedikit ke kanan dan dengan ekspresi tersenyum . Mendapatkan hasil yang sesuai dengan toleransi 0,34 & 0,99.
5			Face 0,24 Neutral 0,99	Posisi pengambilan foto hadap sedikit ke kiri dan dengan ekspresi datar. Mendapatkan hasil yang sesuai dengan toleransi 0,24 & 0,99.
6			Face 0,31 Happy 0,99	Posisi pengambilan foto hadap sedikit ke kiri dan dengan ekspresi tersenyum. Mendapatkan hasil yang sesuai dengan toleransi 0,31 & 0,99.
7			Face 0,32 Neutral 0,99	Posisi pengambilan foto hadap sedikit ke atas dan dengan ekspresi datar. Mendapatkan hasil yang sesuai dengan toleransi 0,32 & 0,99.

8			Face 0,29 Happy 0.99	Posisi pengambilan foto hadap sedikit ke atas dan dengan ekspresi tersenyum. Mendapatkan hasil yang sesuai dengan toleransi 0,29 & 0,99
9			Face 0,28 Neutral 0.99	Posisi pengambilan foto hadap sedikit ke bawah dan dengan ekspresi datar. Mendapatkan hasil yang sesuai dengan toleransi 0,28 & 0,99.
10			Face 0,27 Happy 0.99	Posisi pengambilan foto hadap sedikit ke bawah dan dengan ekspresi datar. Mendapatkan hasil yang sesuai dengan toleransi 0,27 & 0,99.
11	No Train Data		Not Detected	Tidak terindikasi dan tidak ada kesesuaian citra wajah, meskipun dengan orang yang sama, terjadi karena tidak ada <i>Training data</i> yang sesuai dengan posisi wajah tersebut.
12	No Train Data		Not Detected	Tidak terindikasi dan tidak ada kesesuaian citra wajah, meskipun dengan orang yang sama, dan foto hampir sama dengan <i>Training data</i> namun ada beberapa pixel wajah yang tertutup, sehingga tidak dapat dikenali.

Dari tabel tersebut menggambarkan hasil *face recognition* dari aplikasi yang digunakan, hasil perhitungan dari metode CNN dengan hasil data percobaan yang dilakukan didapatkan hasil yang sama dengan. Pada proses perhitungan menggunakan foto pada tabel nomor 2. Hasil perhitungan tingkat *error rate* menunjukkan hasil 0.26 pada hasil *real* dengan menggunakan aplikasi didapatkan hasil 0.26 dari hasil tersebut terlihat sama dengan hasil perhitungan, namun disuatu kondisi tertentu hasil antara perhitungan matematik dan hasil percobaan menggunakan aplikasi terdapat selisih antara 0.1 – 0.6 dikarenakan faktor seperti pencahayaan kurang, kualitas kamera, tingkat kecepatan komputer dalam melakukan penangkapan gambar dan lain-lain, faktor tersebut tentu sangat mempengaruhi hasil. Pada tabel nomor 13 merupakan model data yang sesuai dengan aturan yang berlaku dan posisi dan pelaku sesuai dengan *data training*, namun aplikasi tidak dapat melakukan proses pengklasifikasian karena ada beberapa citra wajah yang tertutup yaitu hidung dan mulut. Pada proses CNN melakukan konvolusi dari setiap pixels yang ada, jika tertutup oleh sesuatu, maka dapat mempengaruhi hasil. Pada gambar nomor 14 merupakan percobaan dengan wajah yang tidak dilakukan proses *training data* dan dengan model posisi wajah sesuai aturan yang berlaku, namun sistem tidak merespon dan memberikan prediksi, karena tidak ada data yang sesuai dengan model wajah tersebut.

Dari hasil *output face recognition* menggunakan aplikasi tersebut, akan diintegrasikan dengan simulasi *barrier gate system* dalam proses Iot. Tahap awal yang harus dilakukan yaitu melakukan pengkodean terhadap *NodeMCU* untuk melakukan pengaturan *network* yang akan digunakan. Pengkodean *NodeMCU* menggunakan *software Arduino.IDE*, berikut penggambarannya.



Gambar 4. 32 *Face Recognition Happy And Open.*

Pada hasil percobaan pada gambar, didapatkan hasil wajah terdeteksi sebagai langgeng dengan *error* sebesar 0.26 dengan ekspresi *happy* dengan toleransi angka 0.98. Dari model hasil *output* tersebut *hardware* dapat merespon dengan membuka, dapat terjadi karena *variable* yang dibutuhkan hardware terkondisi *true*.

Kelebihan dari metode CNN dapat mengidentifikasi wajah dengan berbagai model posisi wajah, dengan aturan sesuai dengan *training data* yang sudah dilakukan, berikut penggambaran model posisi wajah yang berbeda dari posisi wajah sebelumnya.



Gambar 4. 33 *Face Recognition Neutral Side And Close.*

Pada percobaan tersebut dengan model posisi wajah menghadap kanan dari penguji didapatkan hasil *output face recognition* sebagai “Langgeng04” dengan

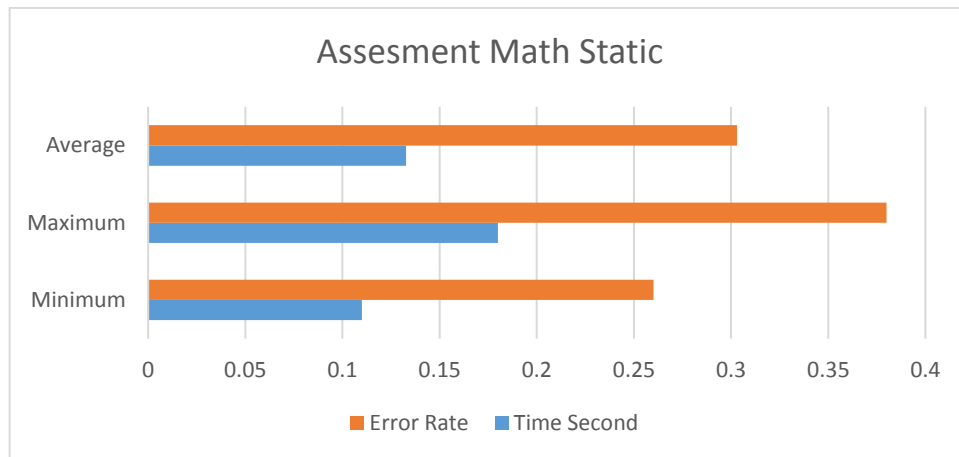
Grafik *line* tersebut menggambarkan hasil pengujian *match static* oleh 100 data uji. Pada grafik mendapatkan kesimpulan bahwa waktu tercepat yang diperoleh yaitu 0.11 *second* dan nilai tingkat *error* paling rendah yaitu 0.26. untuk waktu terlama yang diperoleh yaitu 0.18 *second* dan nilai *error* paling tinggi sebesar 0.38.

Hasil yang diperoleh oleh aplikasi dari uji performa dan uji tingkat *error* sangat bergantung dengan kondisi lapangan, banyak faktor yang mendukung dari hasil pengujian, seperti kualitas kamera, intensitas cahaya, performa *device* yang digunakan. Dari hasil pengujian *match static* dapat diperoleh hasil penilaian berupa rata rata dari *maximum*, *minimum*, dan *average* dari total data uji yang dilakukan yang disajikan pada tabel berikut.

Tabel 4.4 Assessment Pengujian Math Static

Assessment	Time Second	Error Rate
Minimum	0.11	0.26
Maximum	0.18	0.38
Average	0.1327	0.3031

Dari tabel penilaian di atas, dapat diambil kesimpulan bahwa dari total data keseluruhan aplikasi dapat mengenali sebuah citra inputan dalam waktu yang paling cepat yaitu 0.11 *second* dengan tingkat *error* paling rendah berkisaran 0.26. Kemudian tingkat waktu yang paling lama yang dilakukan oleh aplikasi dalam mengenali sebuah citra inputan yaitu berkisar 0.18 *second* dengan tingkat *error* paling tinggi sebesar 0.38. Kemudian rata rata dari waktu tercepat dan tingkat *error* oleh aplikasi yaitu 0.1327 *second* dan tingkat *error* berkisaran 0.3031. Penggambaran grafik hasil penilaian dapat dilihat pada gambar berikut.



Gambar 4. 36 Grafik *Assessment Math Static*

Dari gambar grafik diatas, merupakan penggambaran hasil penilaian aplikasi menggunakan pengujian *math static* yang digambarkan dengan model diagram *line*. Garis berwarna jingga menggambarkan tingkat *error* dan garis berwarna biru menggambarkan *time* dengan satuan *second*. Pengujian berikutnya yaitu dengan teknik pengujian ROC dimana pemfokusan pengujian pada tingkat keakuratan sistem dalam mengenali segala data citra wajah baik yang sudah dilakukan proses *training* maupun belum dilakuan proses *training*. Skenario pengujian seperti pada gambar tabel berikut.

Tabel 4.5 Skenario Pengujian ROC

No	Testing Model	Training Data	Amount Data	Testing Data	Amount data
1	Face Recognition	100%	100	100% Trained	100
2	Face Expression	100%	100	100% Trained	100
3	Face Recognition	100%	100	100% Not Trained	100
4	Face Expression	100%	100	100% Not Trained	100

ROC True Value

TPR TNR FPR FNR TPR TNR FPR FNR

Face Identification Face Expression

Langgeng Sinta Fanggy Arif Firdaus Akbar Adi Catur Faruq Yudha

agram tersebut menggambarkan hasil pengujian ROC pada *true* arti pengujian dengan data yang sudah dilatih. Pada tabel 4.6 model ini jumlah nilai angka total 95 data yang diprediksi benar pada *face* an total 96 data yang dikatakan benar pada *face expression*. Dari apat dilakukan proses perhitungan untuk menentukan *accuracy* dan ankan persamaan sebagai berikut.

$$Error\ Rate = \left(\frac{(FPR + TNR)}{\sum Data} \right) \times 100\% \quad (25)$$

$\Sigma Data: \text{Total Test Data}$

Dari persamaan tersebut dalam menentukan nilai akurasi sistem dan *error* pada proses *face identification* yang dihasilkan sebagai berikut.

$$Accuracy = \left(\frac{(TPR + FNR)}{\sum Data} \right) \times 100\%$$

$$Accuracy = \left(\frac{(95 + 0)}{100} \right) \times 100\%$$

$$Accuracy = \left(\frac{95}{100} \right) \times 100\%$$

Accuracy = 95 %

Hasil akurasi aplikasi dari pengujian ROC pada data yang sudah dilatih mendapatkan akurasi sebesar 95 %. Kemudian melakukan pengujian *error rate* dengan menggunakan persamaan sebagai berikut.

$$Error\ Rate = \left(\frac{(FPR + TNR)}{\sum Data} \right) \times 100\%$$

$$Error\ Rate = \left(\frac{(4 + 1)}{100} \right) \times 100\%$$

$$Error\ Rate = \left(\frac{5}{100}\right) \times 100\%$$

Error Rate = 5 %

Hasil tingkat *error* aplikasi dari pengujian ROC pada data yang sudah dilatih mendapatkan *error* sebesar 5 %. Proses selanjutnya yaitu meletakkan hasil pada tabel model pengujian ROC sebagai berikut.

Tabel 4.7 Hasil ROC True Condition Face Identification

Face Identification		
	Actual Condition Positive	Actual Condition Negative
Predicted Condition Positive	95	1
Predicted Condition Negative	4	0
Accuracy	95%	
Error Rate	5%	

Tabel tersebut merupakan tabel pelatikan hasil perhitungan untuk menentukan akurasi sistem dan tingkat *error* sistem yaitu sebesar 95% untuk akurasi dan 5% untuk tingkat *error*. Kemudian melakukan pengujian pada *face expression* pada model pengujian ROC dengan proses perhitungan sebagai berikut.

$$Accuracy = \left(\frac{(TPR + FNR)}{\sum Data} \right) \times 100\%$$

$$Accuracy = \left(\frac{(96 + 0)}{100} \right) \times 100\%$$

$$Accuracy = \left(\frac{96}{100} \right) \times 100\%$$

Accuracy = 96 %

Hasil akurasi aplikasi dari pengujian ROC pada data yang sudah dilatih mendapatkan akurasi sebesar 96 %. Kemudian melakukan pengujian *error rate* dengan menggunakan persamaan sebagai berikut.

$$Error\ Rate = \left(\frac{(FPR + TNR)}{\sum Data} \right) \times 100\%$$

$$Error\ Rate = \left(\frac{(4 + 0)}{100} \right) \times 100\%$$

$$Error\ Rate = \left(\frac{4}{100}\right) \times 100\%$$

Error Rate = 4 %

Tabel 4.8 Hasil ROC True Condition Face Expression

Tabel tersebut merupakan tabel pelatakan hasil perhitungan untuk menentukan akurasi sistem dan tingkat *error* sistem yaitu sebesar 96% untuk akurasi dan 4% untuk tingkat *error*. Kemudian melakukan pengujian dengan data yang belum dilatih dengan proses yang sama pengujian sebelumnya, tabel pengujian ROC dengan data yang belum dilatih sebagai berikut.

No	Testing	Face Identification				Face Expression			
		TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR
1	Dayat	0	0	0	10	0	0	0	10
2	Ainur	0	0	1	9	0	0	0	10
3	Affan	0	0	0	10	0	0	0	10
4	Fajar	0	0	0	10	0	0	0	10
5	Hafidz	0	0	0	10	0	0	0	10
6	Arik	0	0	0	10	0	0	0	10
7	Arjun	0	0	0	10	0	0	0	10
8	Adit	0	0	1	9	0	0	0	10
9	Doni	0	0	0	10	0	0	0	10
10	Dika	0	0	0	10	0	0	0	10
	Total	0	0	2	98	0	0	0	100
	Percent %	0%	0%	2%	98%	0%	0	0%	100%

$$Accuracy = \left(\frac{(TPR + FNR)}{\sum Data} \right) \times 100\%$$

$$Accuracy = \left(\frac{(0 + 100)}{100} \right) \times 100\%$$

$$Accuracy = \left(\frac{100}{100} \right) \times 100\%$$

Hasil akurasi aplikasi dari pengujian ROC pada data yang tidak dilatih mendapatkan akurasi sebesar 100%. Kemudian melakukan pengujian *error rate* dengan menggunakan persamaan sebagai berikut.

$$Error\ Rate = \left(\frac{(FPR + TNR)}{\sum Data} \right) \times 100\%$$

$$Error\ Rate = \left(\frac{(0 + 0)}{100} \right) \times 100\%$$

$$\text{Error Rate} = \left(\frac{0}{100}\right) \times 100\%$$

Hasil tingkat *error* aplikasi dari pengujian ROC pada data yang sudah dilatih mendapatkan *error* sebesar 0%. Proses selanjutnya yaitu meletakkan hasil pada tabel model pengujian ROC sebagai berikut.

Tabel 4.11 Hasil ROC False Condition Face Expression

Face Expression		
	Actual Condition Positive	Actual Condition Negative
Predicted Condition Positive	0	0
Predicted Condition Negative	0	100
Accuracy	0%	
Error Rate	100%	

Tabel tersebut merupakan tabel pelatikan hasil perhitungan untuk menentukan akurasi sistem dan tingkat *error* sistem yaitu sebesar 100% untuk akurasi dan 0% untuk tingkat *error*. Kesimpulan yang didapat bahwa untuk mendeteksi *face expression* pada data yang belum dilatih mempunyai tingkat akurasi yang besar yaitu 100% dan dengan *error* 0%.

Semua bentuk pengujian yang sudah dilakukan mempunyai banyak faktor yang mempengaruhi dari setiap *value* yang dihasilkan. Faktor tersebut dapat berupa seperti kualitas kamera, intensitas cahaya, performa *device* yang digunakan dan banyaknya jumlah data latih yang digunakan.

4.7. Evaluasi Hasil

Evaluasi hasil merupakan proses percobaan dari keseluruhan model dari mendeteksi citra inputan secara *real-time* dari *actual condition*, hasil *output face recognition* berupa *result condition*, *error rate*, persentase keberhasilan per label dan *response time* yang berupa seberapa cepat hasil *result condition* diukur menggunakan *millisecond* yang diterima oleh *nodeMCU*. Setelah dilakukan proses pengujian, ternyata hasil pengujian antara perlabel, maupun perdata gambar, hasilnya tidak selalu sama, banyak faktor yang menunjang seperti kecepatan gelombang, intensitas cahaya, performa PC yang digunakan. Semua hasil percobaan akan divisualisasikan pada tabel, kemudian secara keseluruhan hasil akan dirata-rata. Hasil pengujian tersebut dapat dilihat pada tabel berikut.

Tabel 4.12 Evaluasi Hasil Keseluruhan

No	Testing data		Error Rate	Percentage	Response Time (ms)
	Actual Condition	Result Condition			
1	Langgeng	Langgeng10	0.29	100%	0.410888672
2		Langgeng04	0.3		0.602050781
3		Langgeng03	0.26		0.518066406
4		Langgeng07	0.28		0.526855469
5		Langgeng09	0.29		0.423095703
6		Langgeng01	0.28		0.480957031
7		Langgeng05	0.26		0.475878906
8		Langgeng02	0.26		0.470800781
9		Langgeng8	0.3		0.465722656
10		Langgeng06	0.29		0.402350781
11	Rizky	Rizky02	0.3	100%	0.534912109
12		Rizky04	0.32		0.362060547
13		Rizky01	0.3		0.561767578
14		Rizky07	0.3		0.427978516
15		Rizky10	0.31		0.399169922
16		Rizky03	0.29		0.395507813
17		Rizky05	0.3		0.374951172
18		Rizky08	0.28		0.354394532
19		Rizky09	0.32		0.333837891
20		Rizky06	0.3		0.313281251
21	Sinta	Sinta02	0.32	100%	0.466796875
22		Sinta04	0.41		0.430908203
23		Sinta07	0.35		0.398925781
24		Sinta01	0.29		0.62109375
25		Sinta05	0.3		0.532958984
26	Fanggy	Fanggy06	0.32	100%	0.452148438
27		Fanggy01	0.34		0.6015625
28		Fanggy03	0.31		0.445068359
29		Fanggy04	0.3		0.504150391
30		Fanggy09	0.28		0.971923828
31	Arif	Arif04	0.38	100%	0.563720703
32		Arif06	0.29		0.58984375
33		Arif01	0.31		0.563964844
34		Arif02	0.29		0.438232422
35		Arif09	0.32		0.784179688
36	Firdaus	Firdaus03	0.29	100%	0.553955078

37		Firdaus04	0.28		0.466064453
38		Firdaus07	0.29		0.624023438
39		Firdaus01	0.34		0.656982422
40		Firdaus02	0.33		0.471923828
41	Akbar	Akbar01	0.29	80%	0.510742188
42		Akbar03	0.28		0.605957031
43		Celvin02	0.7		1.024902344
44		Akbar05	0.3		0.514892578
45		Akbar08	0.28		0.453857422
46	Adi	Adi02	0.32	100%	0.520019531
47		Adi04	0.34		0.611083984
48		Adi07	0.33		0.708251953
49		Adi05	0.33		0.701904297
50		Adi09	0.3		0.507080078
51	Catur	Catur05	0.31	100%	0.536132813
52		Catur09	0.31		0.541992188
53		Catur01	0.29		0.53125
54		Catur08	0.28		0.62890625
55		Catur03	0.3		0.532226563
56	Faruq	Faruq03	0.31	80%	0.550048828
57		Faruq01	0.32		0.628173828
58		Catur04	0.63		1.015136719
59		Faruq08	0.3		0.485107422
60		Faruq10	0.31		0.520019531
61	Dayat	Dayat03	0.29	100%	0.588134766
62		Dayat02	0.28		0.489013672
63		Dayat04	0.3		0.618164063
64		Dayat07	0.29		0.491943359
65		Dayat09	0.3		0.476074219
66	Diana	Diana08	0.28	80%	0.530029297
67		Diana04	0.31		0.423095703
68		Diana07	0.28		0.519042969
69		Sri05	0.65		1.141845703
70		Diana03	0.29		0.590087891
71	Ilham	Ilham02	0.3	100%	0.539550781
72		Ilham04	0.27		0.622070313
73		Ilham05	0.34		0.612060547
74		Ilham08	0.32		0.487304688
75		Ilham01	0.32		0.498046875
76	Ivan	Ivan01	0.28	100%	0.506347656
77		Ivan08	0.3		0.592773438

Untuk melakukan evaluasi akurasi sistem menggunakan pengujian *Confusion Matrix* dengan menggunakan 3 model skenario pengujian berdasarkan tabel pengujian pada ROC yaitu antara *actual condition*, dan *result condition* berikut tabel skenario untuk melakukan evaluasi hasil yang bertujuan untuk mengetahui tingkat akurasi sistem dalam mengenali sebuah citra menggunakan model *confusion matrix* sebagai berikut.

No	Data Latih	Data Uji	Keterangan
1	100%	100%	Evaluasi hasil menggunakan data uji yang sudah dilatih dengan komposisi 100 data latih dan 100 data uji
2	70%	100%	Evaluasi hasil menggunakan data uji yang sudah dilatih dengan komposisi 70 data latih dan 100 data uji
3	30%	100%	Evaluasi hasil menggunakan data uji yang sudah dilatih dengan komposisi 30 data latih dan 100 data uji
4	100%	100%	Evaluasi hasil menggunakan data uji yang tidak dilatih dengan komposisi 100 data latih dan 100 data uji
5	70%	100%	Evaluasi hasil menggunakan data uji yang tidak dilatih dengan komposisi 70 data latih dan 100 data uji
6	30%	100%	Evaluasi hasil menggunakan data uji yang tidak dilatih dengan komposisi 30 data latih dan 100 data uji

Tabel 4.15 Skenario Kedua

Dari hasil pengujian dengan skenario kedua diperoleh hasil jumlah *true positive* (TP) sebanyak 97, *true positive* (TN) sebanyak 0, *true positive* (FP) sebanyak 2, dan *true positive* (FN) sebanyak 1. Maka dapat diperoleh akurasi dengan melakukan perhitungan sebagai berikut:

Accuracy = 97 %

Tabel 4.16 Skenario ketiga

116

Accuracy = 98 %

Tabel 4.18 Skenario Kelima

Dari hasil pengujian dengan skenario kelima diperoleh hasil jumlah *true positive* (TP) sebanyak 0, *true positive* (TN) sebanyak 94, *true positive* (FP) sebanyak 6, dan *true positive* (FN) sebanyak 0. Maka dapat diperoleh akurasi dengan melakukan perhitungan sebagai berikut:

$$Accuracy = \left(\frac{(0 + 94)}{0 + 94 + 6 + 0} \right) \times 100\%$$

Accuracy = 94 %

118

Tabel 4.19 Skenario Keenam

		Predicted Class	
		Positive (P)	Negative (N)
Actual Class	Positive (P)	0 (TP)	8 (FP)
	Negative (N)	0 (FN)	92 (TN)

Dari hasil pengujian dengan skenario keenam diperoleh hasil jumlah *true positive* (TP) sebanyak 0, *true positive* (TN) sebanyak 92, *true positive* (FP) sebanyak 8, dan *true positive* (FN) sebanyak 0. Maka dapat diperoleh akurasi dengan melakukan perhitungan sebagai berikut:

$$Accuracy = \left(\frac{(TP + TN)}{TP + TN + FP + FN} \right) \times 100\%$$

$$Accuracy = \left(\frac{(0 + 92)}{0 + 92 + 8 + 0} \right) \times 100\%$$

$$Accuracy = \left(\frac{(92)}{100} \right) \times 100\%$$

$$Accuracy = 92 \%$$

Maka hasil akurasi yang diperoleh pada skenario keenam yaitu 92%. Semakin tinggi level persentase *accuracy* dari sebuah sistem berarti dikatakan semakin efektif model algoritma klasifikasi yang digunakan dengan pengelompokan presentasi sebagai berikut: (Gorunescu, Florin 2011)

Tabel 4.20 Pengelompokan Persentase Akurasi

No	Low Percentage	High Percentage	Information
1	Akurasi 90 %	Akurasi 100%	Sistem dengan rentan akurasi tersebut dikatakan sangat baik
2	Akurasi 80 %	Akurasi 90 %	Sistem dengan rentan akurasi tersebut dikatakan baik
3	Akurasi 70 %	Akurasi 80 %	Sistem dengan rentan akurasi tersebut dikatakan cukup baik

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian dan pengujian yang sudah dilakukan untuk mengembangkan aplikasi *face recognition* dengan metode *convolutional neural network* untuk simulasi *barrier gate system*, dapat diterapkan sesuai dengan metode dan didapatkan hasil sebagai berikut.

1. Penggunaan metode *convolutional neural network* untuk mengembangkan sistem autentikasi biometrik berbasis *face recognition* untuk simulasi *barrier gate system* dapat dilakukan hingga proses pengujian dan implementasi.
2. Hasil pengukuran kinerja untuk menguji tingkat keberhasilan metode dalam memprediksi hasil *output face recognition* sesuai dengan hasil aktual dengan *time second* paling cepat berkisar 0.11 ms, waktu paling lama berkisar 0.26 ms dan rata rata hasil percobaan secara keseluruhan yaitu 0.1327 ms. Hasil dari pengujian untuk mengukur *error rate* dari aplikasi yang sudah dikembangkan yaitu mendapat *error* paling sedikit berkisar 0.26, untuk tingkat *error* yang paling tinggi 0.38 dan rata rata tingkat *error* yang diperoleh yaitu 0.3031. Evaluasi sistem menggunakan *confusion matrix* untuk mengukur nilai akurasi sistem menggunakan 6 model skenario pengujian mendapatkan akurasi dengan rata rata 93.4%.

DAFTAR PUSTAKA

- Abhirawan, H., Jondri, & Arifianto, A. (2017). Pengenalan Wajah Menggunakan Convolutional Neural Networks (CNN). *Universitas Telkom*, 4(3), 4907–4916.
- Angeline, R., Kavithvajan, K., Balaji, T., Saji, M., & Sushmitha, S. R. (2019). CNN integrated with HOG for efficient face recognition. *International Journal of Recent Technology and Engineering*, 7(6), 1657–1661.
- Athale, S. S., Patil, D., Deshpande, P., & Dandawate, Y. H. (2015). Hardware Implementation of Palm Vein Biometric Modality for Access Control in Multilayered Security System. *Procedia Computer Science*, 58, 492–498. <https://doi.org/10.1016/j.procs.2015.08.013>
- Bah, S. M., & Ming, F. (2019). An improved face recognition algorithm and its application in attendance management system. *Array*, 5(November 2019), 100014. <https://doi.org/10.1016/j.array.2019.100014>
- Coskun, M., Ucar, A., Yildirim, O., & Demir, Y. (2017). Face recognition based on convolutional neural network. *Proceedings of the International Conference on Modern Electrical and Energy Systems, MEES 2017, 2018-Janua*(November), 376–379. <https://doi.org/10.1109/MEES.2017.8248937>
- Dhomne, A., Kumar, R., & Bhan, V. (2018). Gender Recognition Through Face Using Deep Learning. *Procedia Computer Science*, 132, 2–10. <https://doi.org/10.1016/j.procs.2018.05.053>
- Eka Putra, W. S. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1). <https://doi.org/10.12962/j23373539.v5i1.15696>
- Endrianti, F., Setiawan, W., & Wihardi, Y. (2018). Sistem Pencatatan Kehadiran Otomatis di Ruang Kelas Berbasis Pengenalan Wajah Menggunakan Metode Convolutional Neural Network (CNN). *JATIKOM - Jurnal Aplikasi Dan Teori Ilmu Komputer*, 1(1), 40–44.
- Fahmi, R. (2018). *TensorFlow.js Tutorial*. <https://medium.com/@rizafahmi22/screencast-singkat-tentang-tensorflow-js-7e7c3aa6506e>
- Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). Computer vision algorithms and hardware implementations : A survey. *Integration, the VLSI Journal*, 69(August), 309–320. <https://doi.org/10.1016/j.vlsi.2019.07.005>
- Fitriati, D. (2016). Perbandingan Kinerja CNN LeNet 5 Dan Extreme Learning Machine Pada Pengenalan Citra Tulisan Tangan Angka. *Jurnal Teknologi Terpadu*, 2(1), 10–16.
- Gangopadhyay, I. (2018). Face Detection and Recognition Using Haar Classifier and Lbp Histogram. *International Journal of Advanced Research in Computer Science*, 9(2), 592–598. <https://doi.org/10.26483/ijarcs.v9i2.5815>
- Gaouar, L., Benamar, A., & Le, O. (2018). *ScienceDirect HCIDL : Human-computer interface description language for multi-target , multimodal , plastic user interfaces*. 3, 110–130. <https://doi.org/10.1016/j.fcij.2018.02.001>
- Gibert, D., Mateu, C., & Planes, J. (2020). Journal of Network and Computer Applications The rise of machine learning for detection and classification of malware : Research developments , trends and challenges. *Journal of Network and Computer Applications*, 153(January), 102526. <https://doi.org/10.1016/j.jnca.2019.102526>

- Gorunescu, F. (2011). *Data Mining* (1st ed.). Springer-Verlag Berlin Heidelberg. <https://www.springer.com/gp/book/9783642197208#aboutBook>
- Hendri, Z., & Sujana, A. P. (2018). *Sistem Pengenalan Wajah Menggunakan Metode Eigenface Berbasis Raspberry Pi*. 49–57.
- Juneja, K. (2017). MPMFFT based DCA-DBT integrated probabilistic model for face expression classification. *Journal of King Saud University - Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2017.10.006>
- Khotimah, B. K., Sari R, E. M., & Yulianarta, H. (2010). Kinerja metode extreme learning machine (elm) pada sistem peramalan *. *Jurnal Simantec*, 1(3), 186–191.
- Lamani, H. J., Wowor, H., Rumagit, A., & Tuturoong, N. (2012). Implementasi Metode Asynchronous Javascript and Xml (Ajax) Pada Pembuatan Website Universitas Sam Ratulangi. *E-Journal Teknik Elektro Dan Komputer*, 1(1), 1–6.
- Lampropoulos, G., Keramopoulos, E., & Diamantaras, K. (2020). Visual Informatics Enhancing the functionality of augmented reality using deep learning , semantic web and knowledge graphs : A review. *Visual Informatics*, xxxx, 1–11. <https://doi.org/10.1016/j.visinf.2020.01.001>
- Mazur, M. (2015). A Step By Step Backpropagation Example. *Artificial Intelligence*. <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>
- Nalepa, J., Antoniak, M., Myller, M., Ribalta, P., & Marcinkiewicz, M. (2020). *Microprocessors and Microsystems Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation*. 73. <https://doi.org/10.1016/j.micpro.2020.102994>
- O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks*. 1–11. <http://arxiv.org/abs/1511.08458>
- Online, T., Widiyari, C., St, S., Insani, P., Diono, M., & St, S. (2019). *Jurnal Politeknik Caltex Riau Sistem Monitoring Tangki dan Penghitung RunHour Genset Otomatis Berbasis Internet of Things (IoT)*. 5(2), 59–70.
- Phan-Xuan, H., Le-Tien, T., & Nguyen-Tan, S. (2019). FPGA Platform applied for facial expression recognition system using convolutional neural networks. *Procedia Computer Science*, 151(2018), 651–658. <https://doi.org/10.1016/j.procs.2019.04.087>
- Putro, M Dwisnanto, dkk. (2012). Sistem Deteksi Wajah dengan Menggunakan Metode Viola-Jones. *Seminar Nasional "Science, Engineering and Technology,"* 1–5.
- Revina, I. M., & Emmanuel, W. R. S. (2018). A Survey on Human Face Expression Recognition Techniques. *Journal of King Saud University - Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2018.09.002>
- Saha, S. D. T. S. (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- Santhoshkumar, R., & Kalaiselvi Geetha, M. (2019). Deep learning approach for emotion recognition from human body movements with feedforward deep convolution neural networks. *Procedia Computer Science*, 152, 158–165. <https://doi.org/10.1016/j.procs.2019.05.038>
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 18(01), 15–21.

<https://doi.org/10.23917/emitov.v18i01.6235>

- Schrouff, J., Stiffler, R., Bertocci, M., Bebko, G., Chase, H., Lockovitch, J., Aslam, H., Graur, S., Greenberg, T., Pereira, M., Oliveira, L., Phillips, M., & Mourão-miranda, J. (2019). *NeuroImage : Clinical Predicting anxiety from wholebrain activity patterns to emotional faces in young adults : a machine learning approach*. 23(March). <https://doi.org/10.1016/j.nicl.2019.101813>
- Septianto, T., Setyati, E., & Santoso, J. (2018). Model CNN LeNet dalam Rekognisi Angka Tahun pada Prasasti Peninggalan Kerajaan Majapahit. *Jurnal Teknologi Dan Sistem Komputer*, 6(3), 106–109. <https://doi.org/10.14710/jtsiskom.6.3.2018.106-109>
- Setiadi, H., Priyandari, Y., & Cahyono, S. I. (2017). Implementation of Parking System Based on Radio Frequency Identification (RFID) at the Faculty of Engineering Sebelas. *ITSMART: Jurnal Ilmiah Teknologi Dan Informasi*, 6(1), 39–44.
- Shustanov, A., & Yakimov, P. (2017). ScienceDirect ScienceDirect CNN Design for Real-Time Traffic Sign Recognition. *Procedia Engineering*, 201, 718–725. <https://doi.org/10.1016/j.proeng.2017.09.594>
- Siddharth Das. (2017). *CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more....* November 16. <https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5>
- Suendri. (2018). Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan). *Jurnal Ilmu Komputer Dan Informatika*, 3(1), 1–9.
- Sujatha, B. M., Venukumar, B. V., Madiwalar, C. T., Munna, N. C. A., Babu, K. S., Raja, K. B., & Venugopal, K. R. (2016). Translation Based Face Recognition Using Fusion of LL and SV Coefficients. *Procedia Computer Science*, 89, 877–886. <https://doi.org/10.1016/j.procs.2016.06.077>
- Sun, Y., Xu, C., Li, G., Xu, W., & Kong, J. (2020). Intelligent human computer interaction based on non redundant EMG signal. *Alexandria Engineering Journal*. <https://doi.org/10.1016/j.aej.2020.01.015>
- Suprianto, D. (2013). Sistem Pengenalan Wajah Secara Real-Time. *Sistem Pengenalan Wajah Secara Real-Time Dengan Adaboost, Eigenface PCA & MySQL*, 7(2), 179–184.
- Yang, C., Kim, Y., Ryu, S., & Gu, G. X. (2020). Prediction of composite microstructure stress-strain curves using convolutional neural networks. *Materials and Design*, 189, 108509. <https://doi.org/10.1016/j.matdes.2020.108509>
- Zufar, M., & Setiyono, B. (2016). Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains Dan Seni ITS*, 5(2), 72–77. <https://doi.org/10.12962/j23373520.v5i2.18854>